

УДК 681.518.3

**МОДИФИКАЦИЯ ПРОГРАММНОГО ПАКЕТА SARDANA
ДЛЯ НУЖД СИСТЕМЫ ОКРУЖЕНИЯ ОБРАЗЦА
ЭКСПЕРИМЕНТАЛЬНОЙ СТАНЦИИ 1-1 «МИКРОФОКУС»
ЦКП «СКИФ»**

© А. Н. Скляров^{1,2}, С. А. Веригин^{1,2}, Я. В. Ракшун^{1,3}

¹Институт геологии и минералогии им. В. С. Соболева СО РАН,
630090, г. Новосибирск, просп. Академика Коптюга, 3

²ООО «Микрогеософт»,

630009, г. Новосибирск, ул. Большевикская, 103

³Сибирский государственный университет телекоммуникаций и информатики,
630102, г. Новосибирск, ул. Кирова, 86

E-mail: s.g.u@yandex.ru

Приведены модификации программного пакета Sardana для автоматизации экспериментальных станций синхротронного излучения, созданные для достижения номинальной точности работы позиционеров и меньших временных затрат при минимальном контроле со стороны оператора, а также для выполнения типовых экспериментов и гибкой конфигурации эксперимента. Данные модификации применены в автоматизированной системе управления системой окружения образца экспериментальной станции 1-1 «Микрофокус».

Ключевые слова: программный пакет Sardana, экспериментальные станции синхротронного излучения, автоматизированные системы управления, системы автоматизации в научных исследованиях, многокоординатные системы позиционирования.

DOI: 10.15372/AUT20250204

EDN: DPZYKQ

Введение. Экспериментальная станция 1-1 «Микрофокус» строящегося в Новосибирской области центра коллективного пользования «Сибирский кольцевой источник фотонов» (ЦКП «СКИФ») позволит проводить исследования методами рентгенофлуоресцентного анализа в режимах флуоресцентной томографии и спектросубмикроскопии [1], а также методами монокристалльной и порошковой дифрактометрии в условиях высокого статического давления или высокого статического давления и температуры [2]. Данные методы реализуются с помощью двух блоков системы окружения образца (СОО): конфокальной рентгеновской микроскопии и высокого давления. Каждый из них содержит набор исполнительных механизмов, собранных в логические группы многокоординатных систем позиционирования, которые позволяют с высокой точностью (до 50 нм и 0,02 мрад) перемещать образец для итерационных или непрерывных измерений (поточечного или непрерывного сканирования) в каждой точке N -мерной карты положений исследуемого образца.

К основным программным пакетам для разработки автоматизированной системы управления (АСУ) СОО и её интеграции в общую распределённую систему управления относятся пакеты TANGO [3] и Sardana [4]. Ключевой особенностью комплекса данных программ является самодостаточность системы: их архитектура позволяет работать с установкой на разных уровнях — от контроля одного элемента системы (логика взаимодействия с устройством) до контроля всего комплекса станций (взаимодействие между научными установками, различными системами). Обе системы широко используются в отечественных [5] и западных [6, 7] источниках синхротронного излучения (СИ) и других масштабных научных установках [8].

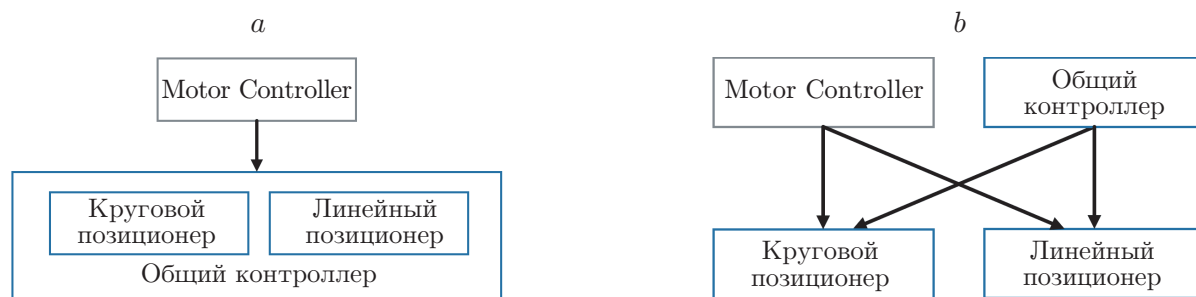


Рис. 1. Архитектура Sardana модулей устройств при использовании общего контроллера: *a* — общий контроллер модуля Sardana, в котором нельзя явно задать разный набор параметров (такая архитектура является классической для модулей Sardana и соответствует паттерну проектирования «Состояние» (State)); *b* — модифицированная архитектура, общий контроллер — дополнительный класс, позволяющий создавать модули устройств с уникальными параметрами без дублирования кода (такая архитектура соответствует паттерну проектирования «Стратегия» (Strategy)). Многоугольниками обозначены различные типы устройств, линиями — наследование. Motor Controller — определённый в класс Sardana

Цель данной работы заключается в модификации программного пакета Sardana для удовлетворения требованиям, предъявляемым к современным экспериментальным станциям источника СИ. Важнейшими из таких требований являются, во-первых, достижение оборудованием своих номинальных характеристик с минимальными требованиями к надлежащему использованию оператором комплекса программ и, во-вторых, обеспечение долгосрочного развития экспериментальной станции как со стороны адаптации АСУ к новому оборудованию, так и со стороны появления новых методов исследования на уже существующем оборудовании.

Программные модули управления движением. Создатели программного пакета Sardana предлагают реализовывать модули управления позиционерами в виде классов, наследующихся от абстрактных классов — контроллеров движения. Подразумевается, что данные классы представляют собой физическое устройство, контролирующее несколько позиционеров (рис. 1, *a*). На экспериментальной станции 1-1 «Микрофокус» ЦКП «СКИФ» используются в том числе универсальные контроллеры для разных типов устройств, поэтому вместо определения классов контроллеров требуется выявить некоторые дополнительные классы, реализующие общие для всех типов приводов функции, а также набор классов Sardana для управления различными типами устройств, которые наследуются от класса Motor Controller и одного из дополнительных классов, как показано на рис. 1, *b*.

В предложенной нами архитектуре (см. рис. 1, *b*) классы определяются не по принципу физических связей с устройствами, как на рис. 1, *a*, а согласно принципу движения, как на рис. 1, *b*. Таким образом, отдельно существуют классы для контроля линейных позиционеров с шаговыми двигателями и угловых позиционеров с двигателями постоянного тока. Это решение позволяет не только выстроить понятную логику разработки программных модулей, но и облегчить дальнейшие модификации системы окружения образца в связи с появлением новых устройств, совместимых с уже существующими универсальными контроллерами.

Алгоритмы высокоточного перемещения. Для некоторых экспериментов критически важна возможность вернуться после сканирования в исходную точку с максимальной точностью. Однако движение элементов, в особенности под нагрузкой, сопряжено с

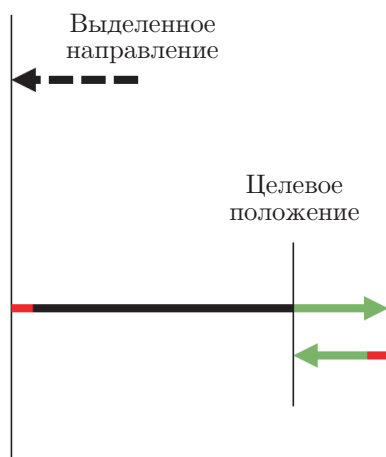


Рис. 2. Схема движения с манёвром для учёта люфта. Зелёным цветом выделены движения манёвра, красным — холостой ход без изменения реального положения

появлением холостого хода при изменении направления движения относительно некоторого выделенного [9]. При этом инкрементальные энкодеры имеют накапливающуюся погрешность измерений при перемещениях на большие расстояния. Заложенные в модули Sardana программные возможности позволяют задавать величину смещения манёвра для учёта этого люфта (Backlash compensation). Схема движения с манёвром для учёта люфта показана на рис. 2.

Для учёта люфта в программном пакете Sardana не предусмотрен выбор выделенного направления движения, относительно которого определяется изменение направления. Кроме того, данную функцию можно отключить, только если задать параметр Backlash, равным нулю. При этом во время проведения эксперимента от оператора требуется контролировать выделенное направление движения и тип перемещения (относительный или абсолютный) для обеспечения оптимальных по точности перемещения и времени исполнения схем перемещения в рамках логики эксперимента. Для реализации данной возможности модули позиционеров расширены четырьмя типами перемещений:

1. Абсолютное — обеспечивает достижение целевой точки на основании показаний энкодеров, а также переопределяет выделенное направление движения, если требуется.
2. Абсолютное высокоточное — обеспечивает достижение целевой точки на основании показаний энкодеров с сохранением выделенного направления движения.
3. Относительное — реализует смещение на заданную величину.
4. Относительное высокоточное — реализует смещение на заданную величину с сохранением выделенного направления движения.

Каждый из типов перемещения предусматривает синхронизацию движения, в том числе для манёвра учёта люфта. Таким образом, надлежащий вызов функций перемещения обеспечивает аппаратную точность позиционирования для комплекса исполнительных механизмов.

Синхронизация движения определяется несколькими способами:

1. Программным — выполняются все возможные действия для подготовки к началу движения, максимально быстро подаётся команда к началу движения всех элементов, участвующих в движении.

2. Программно-аппаратным — элементы, управляемые одним контроллером движения, синхронизируются аппаратно, при этом синхронизация между разными контроллерами осуществляется программным образом.

3. Аппаратным — контроллеры, имеющие необходимые технические возможности, получают аппаратный синхросигнал к началу движения. Элементы, управляемые одним контроллером движения, также синхронизируются аппаратно.

Формирование команд программным пакетом *Sardana* для синхронного движения нескольких позиционеров включает в себя вызов функций подготовки к началу движения каждого из контроллеров, участвующих в движении, с указанием управляемых элементов и целевых положений. Если в случае синхронизации устройств, управляемых одним контроллером, возможно выполнить подготовку к синхронному движению без использования новых подпрограмм, то в случае синхронизации движения с участием нескольких контроллеров требуется создание менеджера движения, который собирал бы все команды подготовки к движению и конфигурировал оборудование соответствующим образом в зависимости от выбранного типа синхронизации. В *Sardana* существует специальный абстрактный класс *trigger/gate*, реализующий данные функциональные возможности, однако, в таком случае потребовалось бы перед каждым экспериментом конфигурировать экземпляр класса *trigger/gate* в соответствии с логикой эксперимента. Поэтому вместо использования стандартных возможностей *Sardana* создан новый класс *Supervisor*, отвечающий за управление конфигурацией движения и синхронизацию между контроллерами. Данный класс является атрибутом модулей *Sardana* и существует в единственном экземпляре на сервере. Его отличие от стандартного класса *trigger/gate* состоит в том, что во время инициализации устройств происходит чтение конфигурации контроллера, в частности определённых в нём логических групп и подключённых устройств. На основании этих данных выделяются ведущие позиционеры группы и задаётся индивидуальная для каждой группы логика синхронизации. Таким образом, конфигурация *Supervisor* происходит автоматически вслед за изменением конфигурации оборудования.

Некоторые контроллеры движения имеют встроенные механизмы для запуска движения по аппаратному синхросигналу, или выдачи синхросигнала в момент старта движения, или достижения постоянной скорости. Конфигурация приёма и передачи таких сигналов в ключевых точках траектории позволяет синхронизировать не только множество подобных контроллеров, но и, что более важно, движение с измерительным комплексом (детекторами). В случае, если несколькими контроллерами движения такая возможность не предусмотрена, команды на старт генерируются заранее и передаются контроллерам.

Кроме того, такая реализация управления синхронизацией движения позволяет формировать команды к исполнительным устройствам, располагая данными о планируемом передвижении и не создавая противоречий между командами управляющей системы. В частности, такой подход даёт возможность получить меньшие временные промежутки между точками во время выполнения макросов непрерывного сканирования, чем позволяют алгоритмы *Sardana*. Работа подобного менеджера движения продемонстрирована в [10, 11] не только для ускорения самих непрерывных сканирований, но и для реализации управления последовательностью импульсов лазера, синхронизированного с работой других модулей.

Макросы. В программном пакете *Sardana* определены макросы (класс *sardana.macros.server.macro.Macro*), представляющие собой алгоритм действий, реализующий эксперимент, его часть или упрощающий взаимодействие оператора с системой. Макросы сканирования — это программный алгоритм действий, определяющий параметры, режимы и порядок работы позиционеров и комплекса измерительных приборов в рамках логики эксперимента.

Согласно документации к программному пакету Sardana, макрос сканирования в конечном счёте вызывает реализацию абстрактного класса сканирования (`sardana.macros.server.scan.gscan.GScan`). Стандартная реализация поточечного сканирования (`sardana.macros.server.scan.gscan.SScan`, потомок класса `GScan`) вызывает работу группы измерительных приборов в каждой точке карты положений. При необходимости выполнения дополнительных действий в точке предусмотрен вызов прикреплённого (hooked) макроса до или после выполнения измерения. Обязательный вызов работы группы измерительных приборов делает невозможными некоторые конфигурации эксперимента, при которых определённые действия в точке требуется выполнить вместо измерения, например пропуск точки по условию или команде оператора.

Для обеспечения такой возможности создан класс `ActionScan` (потомок `sardana.macros.server.scan.gscan.SScan`), вызывающий в точке вместо измерения специальный макрос действия в точке (`Action`). Таким образом, алгоритмы обхода точек используются повторно, и разработка большинства новых макросов сводится к созданию нового действия в точке, что достаточно просто сделать без специальных знаний.

Макросы, разработанные в данной архитектуре, представляют собой библиотеку модулей Python (`MacroScan`). Они учитывают наличие изменений, связанных с алгоритмами высокоточных перемещений в модулях, и формируют алгоритм обхода точек, задавая неуточнённые параметры таким образом, что, во-первых, достигается аппаратная точность перемещения, если это возможно, а во-вторых, минимизируется время обхода всех точек карты положений. По окончании движения все изменённые параметры участвовавших в перемещении элементов возвращаются к исходным. От оператора в этом случае не требуется контроль за введёнными ранее параметрами при использовании таких макросов.

Уменьшение времени проведения эксперимента за счёт алгоритма перемещений достигается следующим образом:

1. В отличие от стандартного манёвра для учёта люфта, используемого Sardana в каждой точке, манёвр выполняется только при изменении направления движения и только актуаторами, направление движения которых изменилось.

2. Выбор направления сканирования по каждой из осей производится таким образом, чтобы количество манёвров для учёта люфта было минимальным. Это не только экономит время, но и уменьшает количество действий, которые так или иначе вносят некоторую погрешность позиционирования.

3. При двумерном сканировании (в том числе в составе трёхмерного) в качестве первой оси сканирования выбирается более длинная либо состоящая из меньшего количества точек, чтобы минимизировать перемещения на большие расстояния и количество манёвров для учёта люфта.

4. Для всех двумерных сканирований реализован «быстрый режим». При его использовании не применяется манёвр для учёта люфта, а обход точек производится по траектории «змейка», что уменьшает средние расстояния передвижений.

Модули библиотеки `MacroScan`, обеспечивающие функции данных макросов, можно разделить на три вида: 1) функциональные блоки; 2) макросы; 3) действия в точке.

Функциональные блоки, в свою очередь, отвечают за исполнение алгоритмов перемещения и имеют абстрактные методы и базовую реализацию сканирования, а при невозможности корректной работы алгоритмов формируют и вызывают соответствующий набор исключений.

Макросы имеют список параметров эксперимента, его описание и дополнительную информацию для оператора, конфигурируют функциональные блоки в соответствии с логикой конкретного эксперимента, производят анализ списка исключений, возникшего при

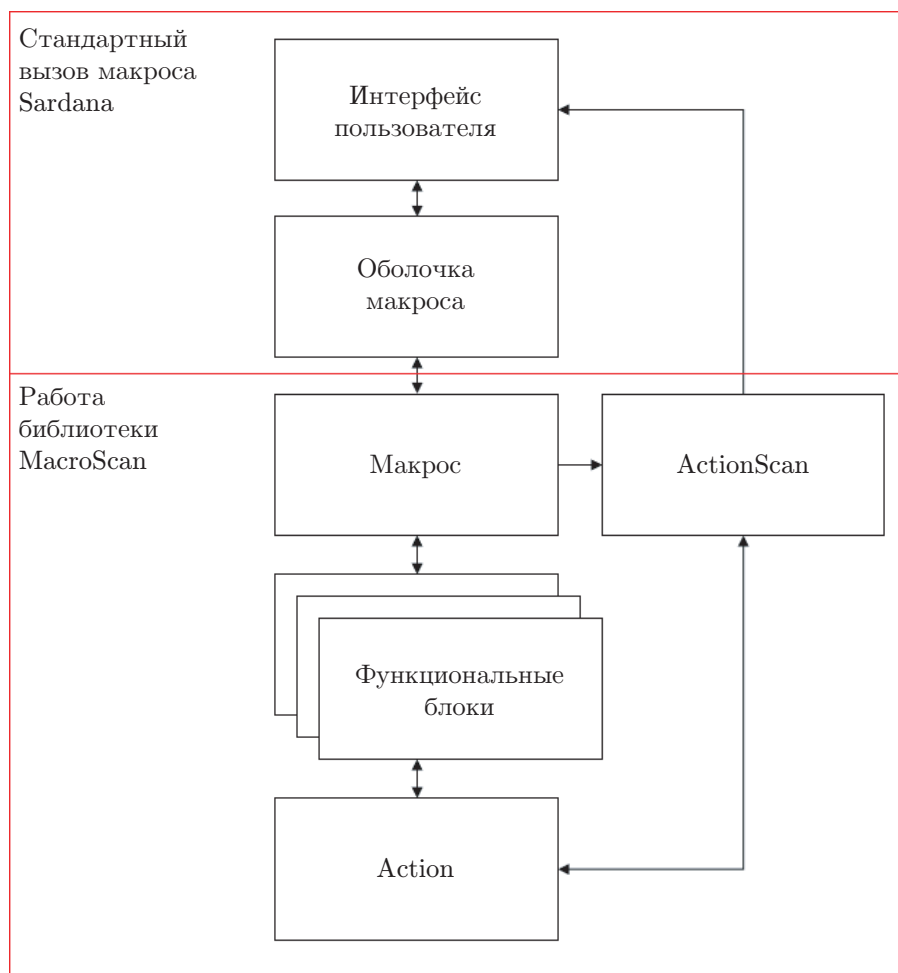


Рис. 3. Иерархия вызовов в макросе сканирования. Стрелки отображают передачу информации

выполнении функциональных блоков, информируют оператора о статусе вызванных операций и в соответствии с настройками вносят изменения в параметры эксперимента или принимают изменения со стороны оператора.

Анализ списка всех возникших исключений на уровне макроса позволяет настроить реакцию на событие и свободно менять регламент работы алгоритма.

Действие в точке отвечает за исполнение операции на карте положений, может быть вызвано как обычный макрос Sardana, предоставляет значения параметров, необходимых для формирования таблицы результатов сканирования, и передаёт их вызвавшему экземпляру ActionScan.

Программный пакет Sardana требует представления библиотеки макросов в виде одного файла со списком классов или функций. Однако, начиная с некоторой сложности архитектуры, необходимы инкапсуляция и разделение классов и функций по пакетам. Для этого макросы, использующие различные функциональные блоки, размещены в разных пакетах. Специальные макросы действия в точке не имеют доступа к пространству имён других пакетов библиотеки. Все макросы имеют доступ только к связанным с их работой модулям. Для удобного применения в интерфейсе Spock используется библиотека макросов, являющаяся оболочкой (оболочка каждого макроса — класс-наследник без пе-

реопределения методов или параметров), которая содержит имена, параметры макросов и информацию для оператора, но для реализации своих функций обращается к библиотеке MacroScan.

Иерархия вызовов представлена на рис. 3. При исполнении любого макроса Sardana первыми вызываются методы Prepare. Для подготовки к выполнению эти методы вызываются в оболочке макроса, затем в макросе, после в функциональных блоках, в конце в Action. Они создают список исключений, который передаётся и обрабатывается в макросе. Далее формируется ActionScan и вызывается метод run, сначала в оболочке, затем в макросе и функциональных блоках. Класс ActionScan вызывает макрос действия Action в каждой точке карты положений.

Реализация разработчиками программного пакета Sardana считывания двумерного канала данных, класса GScan и непрерывных сканирований накладывает некоторые ограничения. Например, невозможно выполнить два непрерывных сканирования в рамках работы одного класса Scan [12].

Для обхода этого ограничения при невозможности использовать ActionScan по выше-названным причинам используется прямое управление модулями устройств для перемещения между точками с последующим вызовом Action в каждой точке.

Заключение. Программный пакет Sardana модифицирован посредством дополнения библиотекой MacroScan. В частности, в ней переработана иерархия вызовов макроса и создан потомок класса SScan для расширения набора действий в точке. Модули Sardana расширены функциональными возможностями для различных типов синхронизации и алгоритмов высокоточных перемещений, которые также интегрированы в новые виды макросов. Всё вышеперечисленное позволило в полной мере реализовать аппаратные возможности СОО экспериментальной станции 1-1 «Микрофокус» ЦКП «СКИФ» для проведения типовых экспериментов в автоматическом режиме, а также упростить дальнейшую модификацию СОО.

СПИСОК ЛИТЕРАТУРЫ

1. Дарьин А. В., Ракшун Я. В. Методика выполнения измерений при проведении рентгенофлуоресцентного анализа с использованием рентгеновской концентрирующей оптики (поликapiлярные линзы) // Научный вестн. НГТУ. 2013. № 2. С. 119–129.
2. Ращенко С. В., Лихачёва А. Ю., Чанышев А. Д., Анчаров А. И. Использование рентгеновской дифракции *in situ* для изучения минеральных превращений: образование лавсонита при 400 °С и 25 кбар // Журнал структурной химии. 2012. **53**, спецвып. 7. С. S46–S49.
3. Götz A., Taurel E., Pons J. L. et al. TANGO a Corba Based Control System // Proc. of the 9th Int. Conf. Accelerator and Large Experimental Physics Control Systems (ICALEPCS). Gyeongju, Korea, 13–17 Oct., 2003. P. 220–222.
4. Reszela Z., Cuní G., Fernandez-Carreiras D. et al. Sardana — a Python Based Software Package for Building Scientific SCADA Applications // Proc. of the 10th Int. Workshop on Personal Computers and Particle Accelerator Controls (PCaPAC). Karlsruhe, Germany, 14–17 Oct., 2014. WCO206. P. 25–27.
5. Peters G. S., Zakharchenko O. A., Konarev P. V. et al. The small-angle X-ray scattering beamline BioMUR at the Kurchatov synchrotron radiation source // Nuclear Instruments and Methods in Phys. Res. Sect. A: Accelerators, Spectrometers, Detectors and Associated Equipment. 2019. **945**. 162616.
6. Liu Z.-K., Chang F.-Y., Chang L.-H. et al. Input Output Controller of Digital Low Level RF System In NSRR // Proc. of the 8th Int. Particle Accelerator Conf. (IPAC). Copenhagen, Denmark, 14–19 May, 2017. THPAB150. P. 4083–4085.

7. **Fernandez-Carreiras D., Becheri F., Blanch S. et al.** Alba, A Tango Based Control System in Python // Proc. of the 12th Int. Conf. Accelerator and Large Experimental Physics Control Systems (ICALEPCS). Kobe, Japan, 12–16 Oct., 2009. THP016. P. 709–711.
8. **Baldini V., Calderone G., Cepparo F. et al.** A complete automatization of an educational observatory at INAF-OATs // Proc. SPIE. 2018. **10707**. 107072C.
9. **Дарьин Ф. А.** Развитие метода конфокальной рентгеновской микроскопии для исследования микровключений в различные геологические матрицы: Дис. ... канд. техн. наук / ИЯФ СО РАН. Новосибирск, 2023. 141 с. EDN: GLBEAG.
10. **Hardion V., Bell P., Eriksson T. et al.** MAX IV Laboratory's Control System Evolution and Future Strategies // Proc. of the 19th Int. Conf. Accelerator and Large Experimental Physics Control Systems (ICALEPCS). Cape Town, South Africa, 9–13 Oct., 2023. TUMBCMO19. P. 395–400.
11. **Bell P., Hardion V., Jamroz J. et al.** Control and Data Acquisition Using TANGO and SARDANA at the Nanomax Beamline at MAX IV // Proc. of the 16th Int. Conf. Accelerator and Large Experimental Physics Control Systems (ICALEPCS). Barcelona, Spain, 8–13 Oct., 2017. TUPHA197. P. 900–904.
12. **Reszela Z., Becheri F., Cuní G. et al.** Sardana Based Continuous Scans at ALBA — Current Status // Proc. of the 16th Int. Conf. Accelerator and Large Experimental Physics Control Systems (ICALEPCS). Barcelona, Spain, 8–13 Oct., 2017. WEBPL06. P. 1067–1072.

Поступила в редакцию 29.10.2024

После доработки 21.12.2024

Принята к публикации 17.03.2025
