

**РАЗРАБОТКА АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ
ДЛЯ ПРОЕКТИРОВАНИЯ ТКАЦКИХ УЗОРОВ
ДЛЯ ТКАЧЕСТВА НА ДОЩЕЧКАХ**

**DEVELOPMENT OF THE APPLICATION ARCHITECTURE
FOR THE DESIGN OF WEAVING PATTERNS FOR WEAVING ON BOARDS**

A.E. МАКАРЕНКО, А.С. ПЕДА

A.E. MAKARENKO, A.S. PEDA

(Государственный университет управления)

(State University of Management)

E-mail: dfkmc@mail.ru; arina.peda01@mail.ru

В статье показана значимость возрождения традиционных ремесел, в частности ручного ткачества на дощечках. Выявлены проблемы, с которыми сталкиваются мастера при проектировании ткацких узоров, и доказана целесообразность автоматизации этого процесса. На основе анализа существующих программных продуктов подтверждена актуальность разработки приложения. Описана технология ткачества на дощечках и сформирована система классов, учитывающая все особенности процесса и являющаяся основой программного приложения, поддерживающего проектирование ткацких узоров и составление инструкций для их воплощения в готовых изделиях. Описанная в статье система классов может служить ядром и при создании приложений, ориентированных на иные техники ткачества.

The article shows the importance of the revival of traditional crafts, in particular, hand weaving on boards. The problems that craftsmen face when designing weaving patterns are identified, and the expediency of automating this process is proved. Based on the analysis of existing software products, the relevance of developing an application is confirmed. The technology of weaving on boards is described and a class system is formed that takes into account all the features of the process and is the basis of a software application that supports the design of weaving patterns and the preparation of instructions for their implementation in finished products. The class system described in the article can also serve as the core when creating applications focused on other weaving techniques.

Ключевые слова: объектно-ориентированное программирование, класс, программное приложение, народный костюм, ткачество на дощечках, ручное ткачество, ткачество поясов, пояс, ткацкий узор.

Keywords: object-oriented programming, class, software application, folk costume, plank weaving, hand weaving, belt weaving, belt, weaving pattern.

Введение

В последние десятилетия практически в каждом уголке России наблюдается рост интереса к национальной истории, жизни и

быту предков, их материальной и духовной культуре. Это касается и традиционного народного костюма, важным элементом которого у многих народностей был

тканый пояс. Наряду с утилитарными, он выполнял ритуальные функции, считался оберегом: узоры костюма выступали "проводниками" человека к высшим силам природы [1...4]. Нарядный пояс являлся украшением, свидетельствовал о благосостоянии владельца [5], демонстрировал мастерство и трудолюбие женщин данной семьи. Выйти на люди без пояса считалось неприличным. Подобное отношение к поясу можно заметить в культуре многих народов России [6...13].

Возрождение традиций ручного ткачества не может не радовать. Люди приобщаются к истории, растет их национальное самосознание, уважение к своим корням, развивается эстетическое чувство, пространственное мышление и творческие способности. Совершенствуя мелкую моторику, рукоделие положительно влияет на когнитивные функции, тренирует память и снижает риск возникновения деменции в пожилом возрасте [14].

Освоив технологию ткачества на готовых схемах, человек испытывает естественную потребность создавать узорные пояса в соответствии с собственным замыслом. На этом этапе он сталкивается с необходимостью отрисовки своих эскизов на бумаге либо при помощи компьютера. И то и другое отнимает много времени, не всегда приводя к желаемому результату. Подобная проблема существует и при проектировании ткацких переплетений для промышленного производства тканей [15...17]. Первые попытки автоматизации проектирования ткацких переплетений относятся к началу 1970-х годов, а стремительное развитие электронно-вычислительной техники на рубеже нынешнего века позволило разработать и внедрить программное обеспечение, позволяющее создавать сложные рисунки, ориентированные на разные виды ткацких переплетений. В монографии [18] подробно рассмотрены вопросы автоматизации проектирования ремизных и жаккардовых тканей, в частности, используемые в этих целях программные продукты.

Ткачество поясов имеет ряд особенностей. Прежде всего, пояс является важным элементом декора костюма, что выдвигает

особые требования к его художественным достоинствам, при этом не только лицевая, но и изнаночная сторона должна выглядеть как минимум аккуратно, не имея длинных перекрытий. Ткань пояса должна быть достаточно толстой и плотной, а также максимально износостойкой, давая возможность завязывать пояс узлом либо фиксировать его застежкой или металлической пряжкой. Ткачество на дощечках позволяет создавать изделия, отвечающие перечисленным требованиям. Однако эта техника отличается от обычного ткачества, поскольку совмещает ткачество с элементами плетения. С усложнением технологии существенно расширяются изобразительные возможности.

Ниша приложений для ткачества на дощечках в настоящий момент практически не заполнена и представлена всего двумя программами для персонального компьютера: GTT [19] и The Loomy Bin [20], причем оба продукта уже не поддерживаются разработчиками. Установка приложения The Loomy Bin вызывает значительные сложности, так как для работы оно требует установленной и настроенной виртуальной машины Java, версия которой не используется и не поддерживается современными операционными системами. Приложение GTT на современных машинах запускается, однако последняя дата в логе изменений приложения GTT – 22.04.2008. Программа GTT позволяет пользователю генерировать схемы для разных видов ткачества, конвертировать схемы и сохранять собственные шаблоны. Однако интерфейс приложения весьма неудобен. Его пестрота и неоднородность мешают работать, панели управления расположены плотно друг к другу, а иконки инструментов не группируются по стилистике или цветовой гамме и практически не несут смысловой нагрузки, что вызывает затруднения при их использовании. В самом приложении отсутствует полная и удобочитаемая справочная система, не реализован поиск по функционалу. Значительным недостатком является невнятная нумерация проходов на схеме, что нередко приводит к ошибкам при ткачестве. В чис-

ле минусов следует назвать и отсутствие русскоязычной версии. На основе проведенного анализа можно утверждать, что существующие решения не отвечают потребностям современного пользователя и актуальной задачей является разработка нового программного продукта.

Методы исследования

Грамотная организация программного продукта обеспечивает удобство работы над проектом, сокращает время разработки, облегчает сопровождение, а часто влияет и на качество готового приложения, например, повышает скорость обработки информации. Программа, назначением которой является создание узоров для традиционного ручного ткачества на дощечках и составление инструкций для воплощения этих узоров в готовых изделиях, ставит перед разработчиком множество нетривиальных задач, для решения которых особенно важно сформировать рациональную систему типов данных и обеспечить их взаимодействие как между собой, так и с инструментами визуального программирования. Объектно-ориентированный подход представляется при этом оптимальным, поскольку позволяет при описании каждой сущности инкапсулировать ее атрибуты и методы работы с ними. Механизмы наследования и агрегирования дают возможность выстроить удобную иерархию, а полиморфизм существенно упрощает работу и сокращает объем кода.

При разработке программного приложения авторы использовали язык программирования C++. Этот язык подразумевается и при написании данной статьи, хотя приводимые ниже соображения могут быть реализованы и посредством иных языков, поддерживающих технологию объектно-ориентированного программирования. Привязка к конкретной среде разработки в статье не делается, поскольку для организации удобного пользовательского интерфейса подойдут многие современные средства визуального программирования. Кроме того, пожелания заказчиков в части интерфейса могут иметь существенные различия, но все они допускают реализа-

цию на основе рассмотренного ниже ядра, представляющего собой систему классов.

Приведем краткое описание сущности процесса ткачества на дощечках.

Дощечка представляет собой тонкую пластину в форме многоугольника (наиболее распространены квадратные дощечки, однако можно ткать и с использованием 3-8-угольных дощечек). В каждом углу имеется отверстие. В отверстия продеваются нити основы. Нити могут быть заправлены как снизу вверх, так и наоборот. Это определяет наклон дощечки в заправке и влияет на вид узора, получаемого при ткачестве. В ходе ткачества наклон дощечки может быть изменен, при этом последовательность нитей меняется на обратную.

Ткачество производится с помощью нескольких дощечек. Комплект дощечек с продетыми в них нитями будем называть заправкой. Число дощечек в заправке может варьировать в широких пределах, чаще всего оно лежит в диапазоне от 8 до 40. Меньшее количество позволяет выполнять лишь простейшие узоры, а при большом количестве затруднительно обеспечить равномерное натяжение долевых нитей (отчасти позволяет решить эту проблему станок). В ходе ткачества две соседние дощечки можно поменять местами, это называется перебросом.

Нити основы со сложенными вместе дощечками равномерно натягивают. Часть нитей при этом оказывается сверху, а часть – снизу: образуется зев, через который пробрасывается уточная нить. После проброса утка дощечки поворачивают, в результате чего формируется узор. Проброс утка и следующий за ним поворот дощечек будем называть проходом. Дощечки можно поворачивать к себе либо от себя. Направление вращения может не совпадать как в разных проходах, так и в рамках одного прохода: это мощное средство для создания интересных и разнообразных орнаментов. Угол поворота чаще всего составляет величину $\pm 360/D$ градусов, где D – количество отверстий дощечки. Для получения повторяющегося узора следует придерживаться определенного алгоритма вра-

щения дощечек. Иногда некорректно выбранная последовательность поворотов чревата слишком длинными протяжками нитей основы. Следует упомянуть технику "холостого хода", при которой на четных проходах поворачивают только четные дощечки, а на нечетных – нечетные. Это вдвое повышает изобразительные возможности заправки, но на выходе получается в два раза более толстое изделие, что желательно далеко не всегда.

При ткачестве на дощечках на каждом проходе можно выделить два активных отверстия. При повороте дощечки от себя (вперед) в узоре появляется цвет нити основы, продетой в первое активное отверстие (обычно верхнее ближнее к ткачу), при повороте к себе (назад) – цвет нити из второго активного отверстия (обычно верхнее дальнее от ткача). При вращении дощечки цвета сменяют друг друга, периодически оказываясь в активных позициях.

Разработка системы классов проводилась снизу вверх. На основе результатов анализа предметной области выполнялось первичное описание класса нижнего уровня, затем осуществлялся переход к классу следующего уровня. В ходе работы над ним, если возникала необходимость, делалась доработка класса нижнего уровня. Возврат к любому из классов допускался при работе над любым из классов более высокого уровня, если того требовала логика разработки. После модернизации класса проводился последовательный анализ связанных с ним классов более высокого уровня, и, если это требовалось, в них вносились соответствующие изменения. В связи с этим особую значимость имеет тщательная проработка классов нижнего уровня.

Общее замечание: при разработке системы классов все их поля объявлялись закрытыми (`private`) либо, в случае наследования, защищенными (`protected`), как это принято в объектно-ориентированном программировании. Методы классов объявлялись общедоступными (`public`).

Результаты и обсуждения

При проектировании снизу-вверх первым следует описать класс нижнего уровня

Дощечка. Поля этого класса, выделенные при первичном описании: количество отверстий, первичный угол наклона и динамический массив цветов нитей. Предполагается, что программа позволит разрабатывать схемы для ткачества с помощью не только квадратных, но и других дощечек. Тщательная и полная проработка методов этого класса существенно упростит моделирование процесса ткачества. Особое внимание надо уделить методу поворота дощечки на заданный угол и методу изменения наклона дощечки, поскольку именно они обеспечивают формирование узора. Важно прописать перегрузки операций, что позволит работать с объектами класса привычным способом. Перегрузка оператора присваивания необходима ввиду того, что одно из полей класса – динамический массив. По этой же причине должны быть явно перегружены конструкторы по умолчанию и копирования и, соответственно, деструктор.

Класс следующего уровня – это Заправка. Поля данного класса при первичном описании: количество дощечек в заправке и динамический массив объектов класса Дощечка. В этом случае используется прием агрегирования, когда некоторый класс содержит в качестве полей объекты, указатели на объекты или массивы объектов ранее описанных классов.

При разработке этого класса возник вопрос: не следует ли включить в состав полей количество отверстий используемых в конкретном случае дощечек? В пользу этого соображения выступает то обстоятельство, что, как правило, все дощечки заправки имеют одинаковое число отверстий (в редких случаях, когда это не так, по числу отверстий отличаются кромочные дощечки, которые в формировании узора участия не принимают). Однако при этом разработчик сталкивается с дублированием данной сущности в классах Заправка и Дощечка, что недопустимо. Между тем, программируя класс Заправка, необходимо постоянно оперировать количеством отверстий. Для решения этой проблемы целесообразно прописать метод, определяющий число отверстий по любой из дощечек заправки.

Аналогичный вопрос возник относительно положения дощечек, то есть того, какие отверстия находятся в активных позициях. На момент начала ткачества положение дощечек одинаково, одинаково оно и при синхронных поворотах. И, в отличие от количества отверстий, дублирования можно было бы избежать, убрав это поле из класса Дощечка. Но синхронное вращение – это частный случай. Более сложные и интересные узоры формируются при асинхронном вращении, поэтому данное поле должно быть привязано к конкретной дощечке и характеризовать ее текущее положение. В связи с этим были внесены изменения в первичное описание класса Дощечка: в состав его полей добавлен признак, характеризующий положение дощечки и определяющий цвета в активных отверстиях.

Класс Заправка содержит в числе полей динамический массив, следовательно, должны быть явно определены конструктор по умолчанию, конструктор копии, деструктор и перегрузка оператора присваивания. Среди методов отметим перестановку соседних дощечек, а также функции сохранения заправки в файл и загрузки ее в программу из файла.

При использовании класса Заправка в ходе разработки программы регулярно возникает необходимость работы с объектами класса Дощечка, входящими в заправку, а именно обращения к их полям. Прямой доступ к ним невозможен, поскольку они являются приватными. Существует два пути решения проблемы. Во-первых, класс Заправка можно объявить дружественным (*friend*) для класса Дощечка. Во-вторых, прописать в классе Дощечка необходимые методы: геттеры и сеттеры. Второй путь предпочтительнее, поскольку он не нарушает закрытости полей, а также более универсален: ведь впоследствии доступ к полям класса Дощечка может потребоваться и из других классов. В связи с этим в описание класса Дощечка внесены изменения: добавлены указанные выше методы.

В класс Заправка также добавлены геттеры и сеттеры, которые с большой долей вероятности потребуются в перспективе.

Класс следующего уровня – это класс Узор. Он фиксирует повороты дощечек на каждом проходе, то есть алгоритм формирования в изделии некоторого рисунка. На практике это обычно формализованное описание одного раппорта. Здесь явно прослеживается необходимость введения низкоуровневых классов, в первую очередь, класса, описывающего единичный акт поворота одной дощечки при одном проходе, назовем его класс Акт.

Класс нижнего уровня Акт включает два поля: угол поворота дощечки и текущий наклон дощечки (как было сказано ранее, в ходе ткачества наклон дощечки может быть изменен). Методы этого класса преимущественно обеспечивают базовую функциональность. Как обычно, должны быть прописаны необходимые перегрузки операций. Класс не содержит указателей и динамических массивов, поэтому в явном описании конструкторов жесткой необходимости нет, однако авторы придерживаются позиции, что конструкторы по умолчанию и копирования лучше определять всегда.

Вернемся к классу Узор. Рисунок раппорта формируется в результате последовательности проходов, а каждый проход – это набор поворотов каждой из дощечек, то есть последовательность объектов класса Акт. В связи с этим возникла мысль: не следует ли ввести класс Проход? Этот класс содержал бы в качестве поля динамический массив объектов класса Акт. Класс Узор содержал бы динамический массив объектов класса Проход либо динамический линейный список объектов этого класса. Второе представляется более интересным решением, поскольку при формировании рисунка в программе заранее неизвестно, сколько проходов в нем будет, они добавляются последовательно. Это позволит удобно фиксировать текущие изменения в объекте. Тем не менее имеет право на существование и второй вариант: формирование рисунка осуществляется в одном из визуальных компонентов, предоставляемых средой разработки; когда пользователь сочтет, что результат удовлетворяет его, то дает команду

"сохранить узор". На этой стадии количество проходов в раппорте уже известно, что позволит выделить в нужном количестве память под динамический массив. Плюсами второго пути является упрощение программного кода и некоторое ускорение работы программы, поскольку сохранение текущего состояния будет осуществляться не постоянно и автоматически, а периодически по желанию пользователя. Существует и третий, альтернативный, путь – двумерный динамический массив объектов класса Акт непосредственно в составе полей класса Узор с периодическим считыванием состояния рисунка по желанию пользователя. Анализ показал, что единственная полезная особенность класса Проход – это возможность использования динамического списка, однако специфические характеристики проходов на данном этапе выявлены не были. Поэтому от введения этого класса в систему решено было отказаться. Таким образом, одно из полей класса Узор – двумерный динамический массив объектов класса Акт. Размерности этого массива – число дощечек и количество проходов, о них говорится ниже.

Нетрудно заметить, что сама по себе инструкция, которая хранится в массиве объектов Акт, информации о цветах и их последовательности не содержит. Но полноценное описание узора должно включать и эти данные, то есть характеристику заправки, на основе которой выполняется узор. Необходимо связать класс Узор с классом Заправка. Какой должна быть эта связь? Вариантов два: либо агрегирование, либо наследование. При первичном описании класса Узор выбрано наследование. Основания для этого были следующие. Прежде всего, это позволит исключить дублирование полей, что существенно упростит процесс программирования и минимизирует возможность конфликта. Также наследование дает возможность реализовать динамический полиморфизм, а именно: если объявить соответствующие методы виртуальными, в ходе работы над рисунком пользователь сможет выбирать как просто заправку, так и узор. При со-

здании класса Узор путем наследования от класса Заправка автоматически решается проблема определения количества дощечек, применяемых для создания рисунка: оно наследуется из родительского класса.

Решение о создании класса Узор путем наследования от класса Заправка потребовало скорректировать последний, а именно определить ряд его методов как виртуальных (virtual).

Для создания двумерного динамического массива объектов класса Акт в классе Узор должно быть предусмотрено также поле, предназначенное для хранения количества проходов. Это целочисленное значение, оно считывается из визуального компонента, в котором отрисован эскиз, когда пользователь обращается к команде "сохранить узор". После этого в необходимом количестве выделяется память под массив и в него считывается последовательность действий (актов), приводящих к получению данного рисунка. Цвета при этом считывать не потребуется, поскольку узор создается на базе уже готовой заправки, полную информацию о которой экземпляр узора унаследовал от класса-родителя.

Класс Узор содержит в числе полей двумерный динамический массив, следовательно, должны быть явно определены конструктор по умолчанию, конструктор копии, деструктор и перегрузка оператора присваивания. Среди методов отметим виртуальные функции сохранения заправки в файл и загрузки ее в программу из файла.

В ходе расширения функционала программы потребовалось внести добавления в первичное описание класса Узор. Дело в том, что в ходе ткачества заправка может претерпевать изменения, например, мастер может изменить наклон тех или иных дощечек либо осуществить их переброс. Все это влияет на результирующий рисунок. Однако первоначальная заправка должна оставаться неизменной, это своего рода точка отсчета, обеспечивающая возможность отката. Если смену направления мы фиксируем в объекте класса Акт, то перестановку дощечек таким образом мы запомнить не можем. Перестановка дощечек соотносится с проходом, в связи с этим

возникла мысль все-таки реализовать этот класс. Но он должен оперировать количеством дощечек, которое уже определено в классе Заправка, причем наследовать класс Проход от класса Заправка не представляется возможным. Это еще раз подтвердило правильность отказа от данного класса. А для фиксации переброса дощечек в классе Узор было предусмотрено специальное поле – строка изменений (либо динамический массив строк изменений по каждому проходу). Парсинг строки позволяет считать информацию о том, в какой момент и какие именно дощечки были переставлены.

Но, наряду с сохранением в "протокол", программа строит рисунок на основе текущего состояния заправки, а оно может быть отлично от первоначального. В связи с этим в класс Узор введено еще одно поле – объект класса Заправка (агрегирование), предназначенное для хранения текущего состояния заправки. На старте (при создании объекта класса Узор) текущая заправка представляет собой копию базовой заправки. Текущая заправка в файл не сохраняется и, соответственно, из файла не считывается. При загрузке Узора из файла она реконструируется по информации, хранимой в строках изменений и в объектах класса Акт.

Разработанная система классов представлена на рис. 1.

Визуализацию процесса ткачества в программе, как и интерфейс, целесообразно реализовать с помощью встроенных визуальных компонентов, предоставляемых средой разработки. Визуальные компоненты и прочие объекты среды построены на основе объектно-ориентированного подхода и в случае необходимости могут быть доработаны с учетом требований к создаваемому программному продукту. Однако основная задача программиста здесь – прописать взаимодействие между разработанными им классами и применяемыми визуальными компонентами. Если упомянутые выше методы классов являются в основном универсальными, то тут потребуются сугубо специфические методы, ориентированные на конкретный инструмент.

```
//-----
class Tabl //Класс Дощечка
{
    // Наклон дощечки
    // Количество отверстий в дощечке
    // Динамический массив цветов (ниток)
    // Активная позиция 1 (ближняя верхняя)
public:
    // Прототипы конструкторов и деструктора
    // Прототипы перегрузок операций и методов
};
//-----
class Zaprav //Класс Заправка
{
protected:
    // Количество дощечек
    // Динамический массив объектов класса Tabl
public:
    // Прототипы конструкторов и деструктора
    // Прототипы перегрузок операций и методов
};
//-----
class Act //Класс Акт
{
    // Угол поворота дощечки
    // Текущий наклон дощечки
public:
    // Прототипы конструкторов
    // Прототипы перегрузок операций и методов
};
//-----
class Uzor: public Zaprav //Класс Узор
{
    // Количество проходов
    // Динамический массив объектов класса Act
    // Динамический массив строк изменений
    // Указатель на объект класса Zaprav
public:
    // Прототипы конструкторов и деструктора
    // Прототипы перегрузок операций и методов
};
//-----
```

Рис. 1

Объявление объектов либо указателей на объекты разработанных классов осуществляется в зависимости от особенностей среды разработки, например, по методу агрегирования в классе, описывающем главную форму проекта.

Заключение

На основе рассмотренной выше системы классов разработана программа, имитирующая процесс ткачества на дощечках с числом отверстий от 2 до 8, позволяющая создавать корректные эскизы и инструкции для их реализации в конкретных изделиях, дающая возможность сохранять результаты работы в файле и считывать их из файла. Программа поддерживает разнообразные техники ткачества на дощечках.

Ручное ткачество на дощечках привлекает людей к истории, воспитывает уважение к своим корням, развивает эстетическое чувство, пространственное мышление и творческие способности, оно прочно заняло место в списке популярных хобби во всем мире.

Приложение будет полезно мастерам, работающим в этой технике, значительно упрощая проектирование ткацких узоров и сокращая временные затраты. Описанная в статье система классов может служить ядром при создании приложений, ориентированных на иные техники ткачества.

ЛИТЕРАТУРА

1. Молчанова Л.А. Удмуртская одежда в народных обрядах // Ежегодник финно-угорских исследований. 2020. Т. 14. № 1. С. 131...137.
2. Баранкевич И.А. Закономерности процесса изготовления костюма в контексте народных представлений восточных славян // Культурная жизнь Юга России. Приложение. 2015. № 1(1). С. 149...153.
3. Третьякова А.Е., Глинкина Е.Т., Сафонов В.В. Исследование и реставрация женской рубахи XIX в. из Тверской губернии // Вестник славянских культур. 2022. № 64. С. 279...291.
4. Зуйкова С.А., Привалов И.В. Значение пояса и платка в русском традиционном костюме: христианская символика и народное осмысление // Культурологический журнал. 2017. №2 (28).
5. Попова Е.А. Фиолетовая гамма в палитре испанского языка: нюансы дополнительной семантики // Вестник Московского государственного лингвистического университета. Гуманитарные науки. 2022. № 3(858). С. 75...82.
6. Корнеева Е.Н. Профессиональные компетенции в обучении художественной обработке текстильных материалов (на материале ручного ткачества) // Наука и школа. 2017. № 6. С. 119...124.
7. Молотова Л.Н., Соснина Н.Н. "Русский народный костюм", в картонном футляре, бумага, печать, "Художник РСФСР", Ленинград, СССР, 1984 г.
8. Мордовский народный костюм: альбом / сост. Т.П. Прокина, М.И. Сурина. Саранск: Мордовское книжное издательство, 1990.
9. Петров И.Г. Чувацкий народный костюм. Рецензия на монографию В.В. Николаева, Г.Н. Иванова-Оркова, В.П. Иванова "Чувацкий костюм от древности до современности" // Вестник Академии наук Республики Башкортостан. 2003. Т. 8. № 2. С. 88...90.
10. Болдырева В.В., Лизина Д.Б. Калмыцкий традиционный женский костюм // Молодой ученый. 2016. № 26 (130). С. 751...753.
11. Султанова А.И. История башкирского национального костюма // Молодой ученый. 2016. № 26 (130). С. 773...775.
12. Печняк В.А. Русский пояс. – URL: https://ruvera.ru/russkij_pojas
13. Молотова Т.Л. Марийский народный костюм. – URL: <https://www.perunica.ru/tradicii/4818-molotova-tl-mariyskiy-narodnyy-kostyum-1992-pdf-rus.html>

14. Панова С.А., Нагаева Е.И., Кириллова А.В., Панов И.Н. Влияние тренировки мелкой моторики на психофизиологические показатели детей с задержкой психического развития // Ученые записки Таврического национального университета имени В.И. Вернадского. Серия: Биология, химия. 2009. Т. 22 (61). № 4. С. 122...127.
15. Ефремов Д.Е., Толубеева Г.И. Теория переплетений. Ч. 1. Главные и производные переплетения. Иваново: ИГТА, 2007.
16. Лунд-Иверсен Б. Ткацкие переплетения / Б. Лунд-Иверсен. М.: Легпромбытиздат, 1987.
17. Цветкова Н.Н. Текстильное материаловедение. СПб.: Изд-во "СПбКО", 2011.
18. Казарновская Г.В., Абрамович Н.А., Самутина Н.Н. Автоматизированные методы проектирования ремизных и жаккардовых тканей. Витебск: ВГТУ, 2014.
19. Guntram's Tablet weaving Thingy – URL: <http://guntram.co.za/tabletweaving/gtt.htm>
20. The Loomy Bin – URL: <http://www.theloomybin.com/index.html>

REFERENCES

1. Molchanova L.A. Udmurt clothes in folk rites / L.A. Molchanova // Yearbook of Finno-Ugric Studies. 2020. Т. 14. № 1. С. 131...137.
2. Barankevich I.A. Laws of the process of manufacture of a suit in a context of national representations of east Slavs / I. A. Barankevich // Cultural life of the South of Russia. Appendix. 2015. № 1(1). С. 149...153.
3. Tretyakova A.E. Research and restoration of a woman's shirt of the XIX V. From the Tver province / A.E. Tretyakova, E.T. Glinkina, V.V. Safonov // Bulletin of Slavic cultures. 2022. № 64. С. 279...291.
4. Zuykova S.A., Privalov I. The meaning of the belt and shawl in the Russian traditional costume: Christian symbolism and folk comprehension // Journal of Culture. 2017. №2 (28).
5. Popova E.A. The purple range in the palette of the Spanish language: nuances of additional semantics / E. A. Popova // Bulletin of the Moscow State Linguistic University. Humanities. 2022. № 3(858). С. 75...82.
6. Korneeva E.N. Professional competences in teaching artistic processing of textile materials (on the material of hand weaving) / E.N. Korneeva // Science and School. 2017. № 6. С. 119...124.
7. Molotova L.N., Sosnina N.N. "Russian folk costume", in a cardboard case, paper, print, "Artist of the RSFSR", Leningrad, USSR, 1984.
8. Mordovian Folk Costume: Album / compilers: T.P. Prokina, M.I. Surina. Saransk: Mordovian Book Publishing House, 1990. 384 с.
9. Petrov I.G. Chuvash folk costume. V.V. Nikolaev, G.N. Ivanov-Orkov and V. P. Ivanov's monograph "Chuvash Costume from Ancient to Modern Times" / I.G. Petrov // Bulletin of the Academy of Sciences of the Republic of Bashkortostan. 2003. Т. 8. № 2. С. 88...90.

10. *Boldyreva V.V.* Kalmyk traditional female costume / V.V. Boldyreva, D.B. Lizinova // Young scientist. 2016. № 26 (130). С. 751...753.
11. *Sultanova A.I.* History of Bashkir national costume / A.I. Sultanova // Young scientist. 2016. № 26 (130). С. 773...775.
12. *Pechnyak V.A.* Russian belt - URL: https://ruvera.ru/russkij_pojas
13. *Molotova T.L.* Mari folk costume - URL: <https://www.perunica.ru/tradicii/4818-molotova-tl-mariyskiy-narodnyy-kostyum-1992-pdf-rus.html>
14. *Panova S.A., Nagaeva E.I., Kirillova A.V., Panov I.N.* The influence of training of fine motor skills on psychophysiological indicators of children with mental retardation / S.A. Panova, E.I. Nagaeva, A.V. Kirillova, I.N. Panov // Scientific Notes of VernadskyTavrida National University. Series: Biology, Chemistry. 2009. Т. 22 (61). № 4. С. 122...127.
15. *Efremov D.E., Tolubeeva G.I.* Theory of weaves. Part 1. Main and derivative weaves. Ivanovo: Ivanovo State Textile Academy, 2007. 33 с.
16. *Lund-Iversen B.* Weaving Interlaces / B. Lund-Iversen. М.: Legprombytizdat, 1987. 104 с.
17. *Tsvetkova N.N.* Textile materials science: Textbook / N.N. Tsvetkova. St. Petersburg: Publishing House "SPbCO", 2011. 72 с.
18. *Kazarnovskaya G.V.* Automated methods of designing remise and jacquard fabrics: monograph / G.V. Kazarnovskaya, N.A. Abramovich, N.N. Samutina; EE "VGТУ". Vitebsk, 2014. 262 с.
19. Guntram's Tablet weaving Thingy - URL: <http://guntram.co.za/tabletweaving/gtt.htm>
20. The Loomy Bin - URL: <http://www.theloomybin.com/index.html>

Рекомендована кафедрой информационных систем Государственного университета управления. Поступила 26.01.23.