

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
федеральное государственное бюджетное образовательное учреждение высшего  
образования  
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ  
им.В.П.АСТАФЬЕВА  
(КГПУ им.В.П.Астафьева)

Институт/факультет Институт математики, физики и информатики  
(полное наименование института/факультета/филиала)

Выпускающая кафедра Базовая кафедра информатики и  
информационных технологий в образовании  
(полное наименование кафедры)

**Шишкин Владислав Андреевич**

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема СИСТЕМА ДИДАКТИЧЕСКИХ ИГР КАК СРЕДСТВО ФОРМИРОВАНИЯ  
ПРЕДСТАВЛЕНИЙ О ПАРАЛЛЕЛЬНЫХ ПРОЦЕССАХ В ШКОЛЬНОМ  
КУРСЕ ИНФОРМАТИКИ

Направление подготовки 44.04.01 Педагогическое образование  
(код и наименование направления)

Магистерская программа Информатика в образовании  
(наименование программы)

**ДОПУСКАЮ К ЗАЩИТЕ**

Заведующий кафедрой  
д.п.н., профессор Пак Н.И.

(ученая степень, ученое звание, фамилия, инициалы)

16.06.16 г.

(дата, подпись)

Руководитель магистерской программы:  
д.п.н., профессор Пак Н.И.

(ученая степень, ученое звание, фамилия, инициалы)

16.06.16 г.

(дата, подпись)

Научный руководитель: к.п.н., Сокольская  
М.А.

(ученая степень, ученое звание, фамилия, инициалы)

10.06.16

(дата, подпись)

Обучающийся Шишкин В.А.

(фамилия, инициалы)

16.06.2016 г.

(дата, подпись)

Красноярск 2016

## Оглавление

Введение .....	4
ГЛАВА 1. Теоретические основы разработки системы дидактических игр, формирующей первичные представления о параллельных процессах. ....	11
§1. Обоснование необходимости разработки методики обучения основам суперкомпьютерных технологий в школьном курсе информатики .....	11
§2. Роль дидактических игр в формировании параллельного стиля мышления. ....	18
§3. Понятийные проблемы линии алгоритмизации в школьном курсе информатики. ....	35
Выводы по главе 1 .....	46
ГЛАВА 2. Система дидактических игр, как средство формирования первичных представлений учащихся о параллельных процессах. ....	48
§1. Дидактические принципы, цель и содержание системы игр .....	48
§2. Типы дидактических игр, формирующих представления о параллельных процессах и их место в курсе информатики для 1-6 классов. ....	60
Выводы по главе 2 .....	78
Заключение .....	79
Библиографический список .....	80
ПРИЛОЖЕНИЕ 1 .....	86
ПРИЛОЖЕНИЕ 2 .....	90
ПРИЛОЖЕНИЕ 3 .....	91
ПРИЛОЖЕНИЕ 4 .....	92

ПРИЛОЖЕНИЕ 5.....	96
ПРИЛОЖЕНИЕ 6.....	97
ПРИЛОЖЕНИЕ 7.....	98
ПРИЛОЖЕНИЕ 8.....	100
ПРИЛОЖЕНИЕ 9.....	102
ПРИЛОЖЕНИЕ 10.....	104
ПРИЛОЖЕНИЕ 11.....	108
ПРИЛОЖЕНИЕ 12.....	111
ПРИЛОЖЕНИЕ 13.....	115
ПРИЛОЖЕНИЕ 14.....	117

## Введение

В настоящее время наблюдается стремительный прогресс в области вычислительной техники и программного обеспечения. Многопроцессорные вычислительные системы начинают активно внедряться во все сферы научной и производственной деятельности информационного общества. Меняются вычислительные методы решения задач, совершенствуются технологии программирования.

Параллельные вычисления и параллельное программирование постепенно становятся необходимыми разделами в содержании дисциплин вычислительной математики и программирования.

Эффективное обучение программированию является существенной проблемой не только для педагогических университетов. Несмотря на разнообразие дидактических методов и приемов преподавания материала, усвоение знаний при изучении новых языков программирования в подавляющем большинстве случаев не достигает требуемого уровня. История развития программирования показывает, что лишь небольшой процент людей разбирается в нем на достаточно высоком уровне. Текущий этап развития программирования характеризуется широким распространением и постоянным совершенствованием технологий параллельного программирования, что связано как с изменением архитектуры вычислительных систем, так и с ужесточением требований к программному обеспечению и к точности научных расчётов. Современные кристаллы процессоров содержат несколько ядер, а в типовых серверах устанавливается несколько процессоров. Стремительно развивается использование серверных кластеров и систем распределенных вычислений. Одним словом, параллельные вычисления становятся неотъемлемой частью нашей жизни.

Технология параллельного программирования существенно изменяет алгоритмическую деятельность программиста, внося в нее новые этапы и

действия. При этом программирование с помощью параллельной технологии меняет не только способы построения программы, но и мыслительную деятельность человека, формирует специфический стиль мышления - параллельный. С другой стороны, способность человека приобрести умения осуществлять параллельную обработку информации и разрабатывать алгоритмы и программы для суперкомпьютерных систем возможна лишь при условии сформированного параллельного стиля мышления.

Таким образом, требования общества к развитию способностей человека осуществлять обработку больших объемов информации с помощью современных суперкомпьютерных технологий определяют необходимость еще в школе у учащихся целенаправленно формировать соответствующие мыслительные умения и деятельность на пропедевтическом (интуитивном уровне). Эту задачу может и должен решать учитель информатики.

Для того чтобы подготовить специалистов в области параллельного программирования, необходимо привить навыки решения подобных задач на интуитивном уровне. Нужно проводить пропедевтику знаний по данному направлению, т.е. на интуитивном уровне вводить элементы взаимодействия параллельных процессов на примере их собственной деятельности. Такая пропедевтика может послужить базой для изучения параллельных процессов в вычислительных системах в дальнейшем, а также может служить одной из форм организации коллективной работы по принципу функционирования кластера.

Влияние алгоритмической деятельности на формирование мыслительных операций не вызывает сомнений. Об этом писали выдающиеся ученые Рубинштейн С.Л., Леонтьев А.Н., Гальперин П.Я., Талызина Н.Ф., Беспалько В.П. и др. Опираясь на их труды, можно говорить о том, что разработка параллельных алгоритмов формирует специфические способы мыслительной деятельности. Соответственно, методическая система обучения школьников основам параллельных процессов должна не просто

учитывать это влияние, но и целенаправленно формировать у учащихся соответствующий способ алгоритмической мыслительной деятельности.

Изучение опыта обучения школьников линии «Алгоритмизация и программирование» в базовом курсе школы говорит о недостаточной методической и педагогической проработке данной проблемы. В школьном курсе информатики практически не рассматриваются параллельные процессы и основы суперкомпьютерных технологий.

Массовое распространение параллелизма влечет серьезные последствия, которые еще предстоит выявить и проанализировать. При этом процесс обучения параллельному программированию сталкивается с ещё большими трудностями, чем обучение стандартному, последовательному программированию. Современная теория алгоритмов создавалась в расчете на понятие последовательного алгоритма. В связи с этим возникает ряд вопросов:

1. Как быть с параллельным алгоритмом? Его выполняет один исполнитель или группа исполнителей?

2. Требуется доработки традиционное понятие системы команд исполнителя (СКИ) и само понятие команды. Если мы считаем, что группа исполнителей образует единого исполнителя, то что считать СКИ этого исполнителя? И что считать командой? Или оставить понятие СКИ для каждого отдельного исполнителя? То есть это уже не система команд исполнителя, а система команд одной из компонент исполнителя (для которой еще нет названия)?

3. Что считать параллельным алгоритмом?

4. Среди свойств алгоритмов выделяется новая практически значимая характеристика - способность к распараллеливанию. До какой степени имеет смысл увеличивать количество процессоров при выполнении данного алгоритма?

5. Отдельный вопрос — методы распараллеливания уже существующих последовательных алгоритмов.

Соответственно, возникает необходимость поиска ответов на поставленные вопросы.

Актуальность исследования позволяет выделить следующие **противоречия**:

1. между необходимостью обучать учащихся основам параллельных вычислений и недостаточная готовность учителей проводить такое обучение;

2. между необходимостью формирования представления о параллельных процессах у школьников в курсе информатики и отсутствие методики обучения, формирующей представления о параллельных процессах у школьников;

Выделенные противоречия позволили определить **проблему исследования**: каким образом формировать представления о параллельных процессах в школьном курсе информатики с использованием системы дидактических игр.

Противоречия и проблема позволили выделить объект и предмет исследования.

**Объект исследования**: процесс обучения информатике в 1-6 классах.

**Предмет исследования**: система дидактических игр, формирующая представления о параллельных процессах в 1-6 классах.

Проблема, объект и предмет позволили сформулировать **цель исследования**: теоретически обосновать и разработать систему дидактических игр для формирования первичных (интуитивных) представлений учащихся о параллельных процессах и особенностях совместной (параллельной) работы исполнителей.

В соответствии с выделенными проблемой, объектом, предметом и поставленной целью исследования была сформулирована **гипотеза исследования**.

Формирование первичных представлений учащихся о параллельных процессах будет возможно, если:

1. В процесс изучения информатики в 1-6 классах будет включено предварительно отобранное и адаптированное базовое содержание обучения основам суперкомпьютерных технологий в игровой форме;

2. Определена последовательность изучения базовых понятий и свойств параллельных процессов.

3. Определены игровые методы, приёмы, средства формирования первичных представлений у учеников в области параллельных вычислений;

4. Внедрена в курс информатики система дидактических игр, направленная на последовательное формирование представлений о параллельных процессах и создающая предпосылки к формированию параллельного стиля мышления.

Цель, предмет и гипотеза исследования определили его **ведущие задачи:**

1. Проанализировать литературу по заявленной теме с целью выявления понятийных проблем линии алгоритмизации в школьном курсе информатики.

2. Определить понятие «параллельного стиля мышления», этапы формирования и роль дидактических игр в формировании параллельного стиля мышления.

3. Выделить основные методы, средства и формы, формирующие начальные представления у учеников о параллельных процессах.

4. Уточнить базовое содержание обучения основам суперкомпьютерных технологий и предложить последовательность изучения данной темы.

5. Разработать систему дидактических игр, обеспечивающих формирование первичных представлений учащихся о параллельных процессах.

**Теоретико-методологические основания исследования:** психолого-педагогические особенности обучения детей младшего школьного возраста (Подласый И.П., Коменский Я.А., Ушинский К.Д., Рубинштейн С. Л., Голуб Б.А.); психологические исследования в области влияния деятельности на развитие мышления (Л.С. Выготский, П.Я. Гальперин, А.Н. Леонтьев, С.Л. Рубинштейн); исследования в области параллельных вычислений (Л.В. Городня, Гаврилова И.В., Дитер М.Л., Плаксин М.А., Иванова Н.Г., Кучев А.Д., Сокольская М.А., Турушев М.И., Мэттсон Т., Чернышев А.); исследования в области составления пропедевтических курсов обучения (Л.В. Босова); влияние алгоритмической деятельности на формирование мыслительных операций (Рубинштейн С.Л., Леонтьев А.Н., Гальперин П.Я., Талызина Н.Ф., Беспалько В.П.); исследования в области методов и приемов обучения параллельному программированию (А.В. Козволина, М.В. Слива).

Для решения поставленных задач использовались следующие **методы исследования:** теоретические (изучение и анализ педагогической, психологической, методической и предметной литературы по теме исследования, анализ теоретических и эмпирических данных, изучение и обобщение педагогического опыта, сравнительный анализ).

**Научная новизна** исследования заключается в следующем:

1. Обоснована необходимость формирования у учащихся представлений о параллелизме на интуитивном уровне.
2. Разработана система дидактических игр, нацеленных на формирование у учащихся на интуитивном уровне понятий из области суперкомпьютерных технологий.

**Теоретическая значимость исследования** заключается в:

1. Обосновано теоретическое и практическое содержание вводимых через систему игр понятий.
2. Определено место для изучения параллельных процессов в школьном курсе информатики.

**Практическая значимость исследования** заключается в разработке системы дидактических игр, обеспечивающих формирование представлений учащихся о параллельных процессах.

**Апробация и внедрение результатов.** Материалы исследования были представлены на XXXIII международной научно-практической конференции «Интеллектуальный потенциал XXI века: ступени познания», (Новосибирск, 2016 г.); на XI международной научно-практической конференции «Научные исследования и разработки молодых ученых», (Новосибирск, 2016 г.); на всероссийской (с международным участием) конференции «Актуальные проблемы информатики и информационных технологий в образовании» в рамках XVI международного научно-практического форума студентов, аспирантов и молодых ученых «Молодежь и наука XXI века» (Красноярск 2016 г.).

По теме исследования опубликовано 4 работы. (Приложение 14)

**Структура диссертации.** Работа состоит из 123 страниц, 17 рисунков, 7 таблиц, введения, двух глав, заключения, библиографического списка (51 источника) и 14 приложений.

## **ГЛАВА 1. Теоретические основы разработки системы дидактических игр, формирующей первичные представления о параллельных процессах.**

### **§1. Обоснование необходимости разработки методики обучения основам суперкомпьютерных технологий в школьном курсе информатики.**

В современном обществе высокие информационные технологии стали фундаментальной инфраструктурой, подобно энергетике, дорожным коммуникациям и другим, жизненно важным для экономики государства системам. Научное знание и информация становятся определяющим фактором общественной жизни и производства.

В последнее десятилетие в мире наблюдается огромное увеличение объема информации – каждые три-четыре года он удваивается.

Синхронно с этим процессом в ряде областей науки, техники и управления хозяйственным комплексом появляется все больше задач, которые требуют для своего эффективного решения принципиально новых технологий обработки данных с предельно достижимыми значениями быстродействия средств вычислительной техники.

Передовые информационно-коммуникационные технологии (ИКТ), являются краеугольным камнем развития инфраструктуры современного государства. Нет ни одной отрасли науки, экономики, государственного управления и безопасности, для которых не были бы исключительно важны разработки и реализации, перспективных ИКТ. В первую очередь это относится к суперкомпьютерным технологиям. Без использования современных средств ИКТ эффективное, конкурентоспособное производство становится невозможным ни в одной отрасли [40].

При любом уровне развития компьютерной техники всегда имеются задачи, которые требуют для своего решения использования вычислительной мощности в десятки, сотни или тысячи раз превышающей ту, которая может

быть реализована на одном процессоре. В связи с этим в настоящее время в мире наблюдается своеобразный бум в области стратегически важного направления по созданию высокопроизводительных вычислительных систем с параллельной архитектурой или суперкомпьютеров. Обладание все большими вычислительными мощностями имеет стратегическое значение для развитых государств. В связи с этим, среди ведущих промышленных стран идет острое соперничество за обладание все более совершенными и сверхпроизводительными компьютерными технологиями, как важным стратегическим ресурсом обеспечения развития страны [37].

Практически все развитые страны Запада имеют сегодня национальные программы создания компьютеров сверхвысокой производительности.

Главной целью отрасли параллельных вычислительных технологий обусловлено тем, что десятки, сотни или тысячи современных мощных процессоров, объединенных в единую вычислительную систему, дают новое качество – инструмент обработки информации, мощь которого сравнима с мощью человеческого мозга, а по некоторым параметрам (например, по скорости вычислений) во много раз превосходит его. Такой интеллектуальный ассистент становится незаменимым инструментом или даже мощным оружием, способным дать значительное, если не подавляющее технологическое и военное превосходство над другими государствами.

Мировые тенденции в области разработки и использования технологий высокопроизводительных параллельных вычислений показывают, что задача развития суперкомпьютерных вычислений является актуальнейшей задачей для страны. Экономический прогресс, политическая стабильность и национальная безопасность существенно зависят от успехов в решении данной задачи [39, 41].

Одной из основных проблем в использовании суперкомпьютерных технологий на сегодняшний день является нехватка высокоспециализированных кадров. В связи с этим во многих университетах

создаются лаборатории параллельных вычислительных систем, где обучают студентов правильно использовать оборудование для решения определенных задач. Тем самым пополняется база педагогов, способных обучать детей суперкомпьютерным технологиям, как в основном, так и в дополнительном образовании.

До недавнего времени параллельное программирование было уделом небольшого числа высококвалифицированных системных программистов. Сегодня оно становится частью профессиональной компетенции. Но технология параллельного программирования существенно отличается от традиционного последовательного. В подтверждение этого утверждения вслед за Л.Л. Босовой [4] процитируем крупнейшего российского специалиста в области параллельных вычислений В.В. Воеводина [6, стр.150, 155]:

«... Освоение вычислительной техники параллельной архитектуры ... молодыми специалистами идет с большими трудностями. На наш взгляд, это связано с тем, что знакомство с параллельными вычислениями, как и образование в этой области в целом, начинается не с того, с чего надо бы начинать. К тому же то, с чего надо начинать, не рассказывается ни в каких курсах вообще. Возможность быстрого решения задач на вычислительной технике параллельной архитектуры вынуждает пользователей изменять весь привычный стиль взаимодействия с компьютерами. По сравнению, например, с персональными компьютерами и рабочими станциями меняется практически все: применяются другие языки программирования, видоизменяется большинство алгоритмов, от пользователей требуется предоставление многочисленных нестандартных и трудно добываемых характеристик решаемых задач, интерфейс перестает быть дружественным и т.п. Важным является то обстоятельство, что неполнота учета новых условий работы может в значительной мере снизить эффективность использования новой и, к тому же, достаточно дорогой техники...

Важно лишь, чтобы обучающийся как можно раньше узнал, что существуют другие способы организации вычислительных процессов, а не только последовательное выполнение «операция за операцией», что на этих других способах строится самая мощная современная вычислительная техника, что только на такой технике удастся решать крупные промышленные и научные задачи и т.д. Важно, в первую очередь, для того, чтобы как можно раньше обратить внимание обучающихся на необходимость критического отношения к философии последовательных вычислений. Ведь именно с этой философией им приходится сталкиваться на протяжении всего образования как в школе, так и в вузе. И именно эта философия мешает пониманию особенностей работы на вычислительной технике параллельной архитектуры».

Сегодня нам нужны методики для массового обучения технологии параллельного программирования. В процессе обучения настало время для переворота в отношениях последовательного и параллельного программирования. До сих пор мы сначала учили последовательному программированию, а потом — распараллеливанию последовательных алгоритмов. Сейчас надо ставить вопрос о том, чтобы сразу учить параллельному программированию. А последовательный алгоритм рассматривать как некую часть параллельного алгоритма, которая не требует связи с другими его частями. Как это делать — вопрос открытый.

Тридцать лет назад начинающаяся массовая компьютеризация производства потребовала увеличения уровня компьютерной грамотности населения. Это привело к введению в школьную программу в 1985 г. курса информатики. Но курс информатики в советском (затем в российском) исполнении не сводился к «кнопочной информатике» — к освоению технологии работы с пакетами прикладными программ и компьютерными играми. Он начал изменять стиль мышления подрастающего поколения. В первую очередь это касалось алгоритмичности, точности, строгости. Затем

курс информатики вобрал в себя элементы логики и системного анализа. Впоследствии все это значительно упростило распространение так необходимого в XXI в. проектного подхода. Сейчас речь идет о том, что в течение следующего десятилетия параллельные алгоритмы должны стать элементом общей культуры мышления. Вопрос: каким образом скажется на мышлении следующего поколения освоение понятия параллельного алгоритма, к чему приведет перестройка сознания «на параллельный лад»?

Массовое распространение параллельной обработки информации делает актуальным перемещение соответствующих понятий в разряд общедоступных и общекультурных. Знакомство с параллельными алгоритмами должно стать частью грамотности так, как это за последнюю четверть века произошло с базовыми понятиями теории алгоритмов. Сделать это можно только одним путем — включением соответствующих тем в школьный курс информатики. Значит, нужна методика начального знакомства с параллельным программированием на уровне средней школы.

Исторически первая попытка включения тематики параллельных вычислений в школьный курс информатики была сделана еще двадцать лет назад. Двадцать лет назад в курсе под названием «Алгоритмика» [17, 1] был описан исполнитель «Директор строительства», который командовал параллельными действиями нескольких бригад, строящих сооружение из блоков прямоугольной и треугольной формы. Более того, для этого исполнителя была создана программная реализация. Увы! Эта замечательная методическая разработка в середине 90-х оказалась не востребована. Она почти на двадцать лет опередила свое время!

Сегодня положение сложилось так, что тематика параллельных вычислений в средней школе в первую очередь оказалась связана с темой суперкомпьютеров. Именно на суперкомпьютерах акцентируют внимание учащиеся авторы различных методических разработок [8, 20, 21, 27, 38], даже тогда, когда в этом нет необходимости. Достаточно сказать, что

соответствующий раздел в журнале «Информатика в школе» носит название «Суперкомпьютерное образование в школе». Такая ситуация имеет как положительные, так и отрицательные стороны. Среди положительных сторон надо назвать:

1. Интерес, который вызывает в обществе, в том числе, в среде учащихся, тема суперкомпьютеров. Этот интерес повторяет на современном уровне интерес, который полвека назад вызывали большие машины — суперкомпьютеры своего времени;

2. Организационную поддержку со стороны суперкомпьютерного сообщества. Каждое лето на факультете вычислительной математики и кибернетики МГУ проводится Летняя Суперкомпьютерная Академия [29]. И каждое лето в рамках этой Академии организуется школьный трек для учителей информатики.

3. Интерес и поддержку со стороны российского представительства фирмы Интел и менеджера по стратегическому развитию фирмы Интел Игоря Олеговича Одинцова.

Недостаток «суперкомпьютерного» подхода заключается в зауживании тематики параллельных вычислений. Сами суперкомпьютеры школьникам, как правило, недоступны. Задачи, на решение которых они нацелены, для школьников слишком сложны и, в большинстве случаев, не имеют непосредственной практической значимости и не представляют практического интереса.

Естественным расширением суперкомпьютерной тематики является изучение параллельного программирования. В настоящее время для выполнения параллельных программ совсем не обязательно иметь суперЭВМ. Достаточно многоядерного процессора или видеокарты с набором графических ускорителей. Из работ в этом направлении отметим кандидатскую диссертацию М.А. Сокольской по методике обучения будущих учителей информатики основам параллельного программирования

[39] и опыт Е.Ю. Киселевой по освоению школьниками технологии CUDA[20].

Сосредоточение внимания на суперЭВМ и параллельном программировании существенно обедняет и усложняет тему параллельных вычислений, отвлекает учащихся от множества важных и доступных вопросов. Целью темы «параллельные вычисления» является не обучение «реальному» параллельному программированию (изучение соответствующих языковых конструкций, языков программирования и технологий), а ознакомление учащихся с соответствующим набором понятий и понимание особенностей параллельной работы. Мир вокруг и внутри нас представляет собой сложную параллельную систему. И эта система сама по себе дает массу материала для освоения понятий и механизмов параллелизма. Никакие сложные искусственные конструкции типа технологий MPI и OpenMP для этого не нужны. Школьная информатика должна воспитать мышление, настроенное на «параллельный лад». В школе акцентировать имеет смысл не знакомство с суперкомпьютерами и изучение параллельного программирования, а освоение механизмов «совместной деятельности», постоянно и широко используемых в жизни.

## **§2. Роль дидактических игр в формировании параллельного стиля мышления.**

Основной особенностью информатики является развитие алгоритмического мышления у учащихся посредством обучения основам программирования. Формирование у школьников основ алгоритмического мышления по Г.А. Звенигородскому в 90-е годы прошлого столетия базируется на понятии исполнитель. Исполнитель (алгоритма) – это тот объект, для управления которым составлен алгоритм. Учебные исполнители являются традиционным средством для начального этапа в изучении алгоритмизации и программирования, где существует определенный набор команд и сред, в которых он функционирует. На сегодняшний день используются различные среды, наиболее известные из них: «Кумир», «Роботландия» и «Логомиры» [14, 25].

Но нельзя забывать про актуальность и новизну обучения. За последнее время мир программирования успел значительно измениться, например, CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU (графических процессоров) [48]. Одна из важных проблем в обучении современному программированию – это обучение параллельному программированию, актуальность которого растет с каждым днем из-за перехода на использование многоядерных процессоров и широкое распространение графических процессоров. С помощью параллельного программирования мы разбиваем большие задачи на более мелкие, комплекс которых затем решается одновременно [45].

Для того чтобы подготовить специалистов в области параллельного программирования, необходимо привить навыки решения подобных задач на интуитивном уровне. Нужно проводить пропедевтику знаний по данному направлению.

Согласно большому энциклопедическому словарю понятие ПРОПЕДЕВТИКА (от греч. *propaideuo* - предварительно обучаю) - введение в какую-либо науку, предварительный вводный курс, систематически изложенный в сжатой и элементарной форме [50].

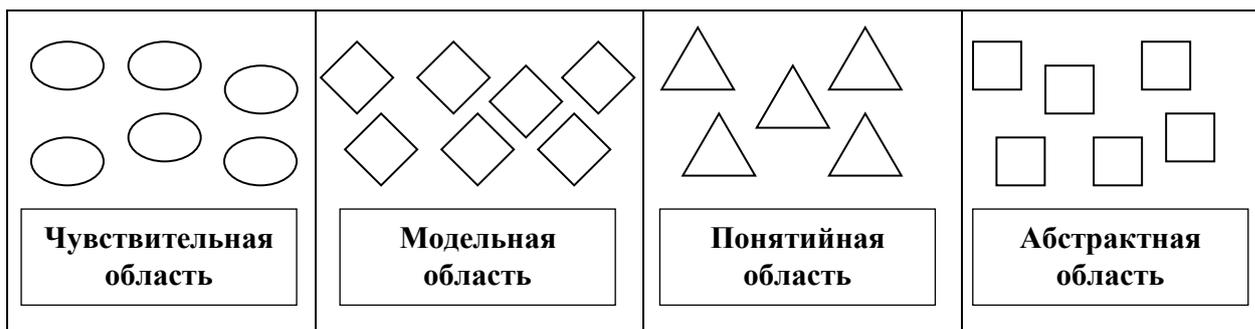
Подходящим возрастом для этого является возраст учащихся младшей и средней школы. Для учеников необходимо проводить пропедевтическую подготовку знаний, т.е. на интуитивном уровне вводить элементы взаимодействия параллельных процессов на примере их собственной деятельности. Такая пропедевтика может послужить базой для изучения параллельных процессов в вычислительных системах в дальнейшем, а также может служить одной из форм организации коллективной работы по принципу функционирования кластера [47].

### **1.2.1. Параллельный стиль мышления и этапы его формирования.**

Формирования представлений о параллельных процессах в школьном курсе информатики может быть успешным при развитии алгоритмического мышления, а также условия изменения характера мыслительной деятельности, формирования специфичного стиля мышления, которое назовём параллельным.

В этой связи необходимо определить, а что же мы будем понимать под «параллельным стилем мышления». Из работ в этом направлении отметим статью Сокольской М.А. и Степановой Т.А. [41], в которой авторы пытаются объяснить на основе сочетания пространственно-временной модели памяти [32] и психологической теории деятельности [26] что же такое параллельный стиль мышления.

Память человека в каждый момент времени может быть условно разделена на 4 области (рис. 1):



*Рис. 1. Пространственно-временная модель структуры памяти*

1. Чувственную область, в которой фиксируются в виде чувственных образов отражения объектов внешнего мира, воспринятые при помощи сенсорной системы;

2. Модельную область, в которой хранятся образы модельного отражения окружающего мира;

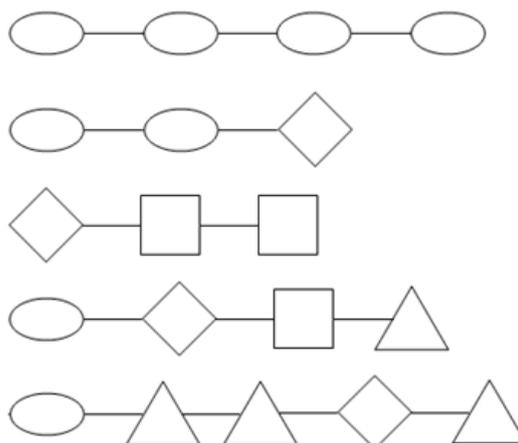
3. Понятийную область, в которой хранятся образы понятий в аудиально-знаковой форме, связанные со свойствами воспринятых объектов и событий [7];

4. Абстрактную область, в которой хранятся образы обобщенных понятий (абстракции).

Извлечение информации из памяти происходит путем активации соответствующих образов в рассмотренных областях. В этой связи можно принять следующие определения базовых понятий:

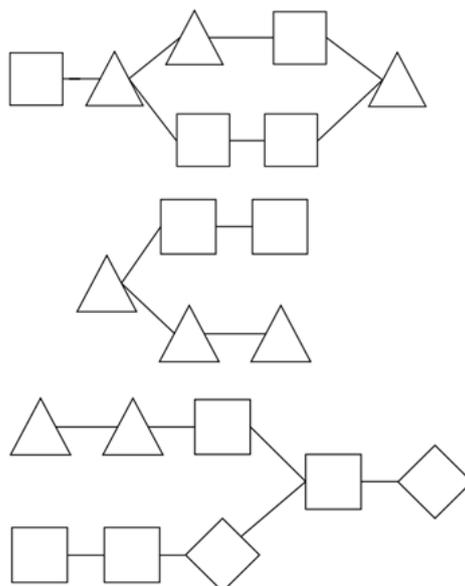
- мысль — это активированный образ в памяти;
- мыслительный процесс, или мышление, — это способ активации последовательной цепочки образов (мыслей).

Цепочки активных образов могут выстраиваться различными способами. Мыслительный процесс может быть чисто чувственным (активная цепочка из образов чувственной области памяти), модельным или абстрактным либо смешанным — активируется последовательность образов из разных областей памяти (рис. 2).



*Рис. 2. Последовательный способ мышления*

Объекты реального мира, воспринимаемые одновременно, образуют в соответствующей области памяти образы, которые также формируются или активизируются одновременно. При этом если объекты реального мира вовлечены в параллельно протекающие несвязанные процессы, то цепочки, порождаемые такими объектами, также возникают параллельно и могут иметь разные конфигурации (рис. 3). С переходом от чувственной и модельной области к понятийной и абстрактной процесс формирования сложных, ветвящихся цепочек, по сути, не изменяется.



*Рис. 3. Параллельный способ мышления*

Согласно положениям психологической теории деятельности А.Н. Леонтьева, формирование рассмотренных на рис. 3 цепочек наиболее эффективно осуществляется в процессе соответствующей деятельности.

Рассмотрим основные положения этой теории. Специальные психические процессы, в том числе и мышление, не могут рассматриваться вне системы отношений деятельность — сознание, их общих системных качеств, поскольку «существуют только в описанных отношениях системы, на тех или иных ее уровнях» [26].

Связи субъекта с материальным миром осуществляются через деятельность, он получает отражение в сознание именно благодаря соответствующей деятельности, поскольку «развитие деятельности необходимо приводит к возникновению психического отражения реальности...» [26].

«Всякий мыслительный процесс является по своему внутреннему строению действием или актом деятельности, направленным на разрешение определенной задачи» [37]. Очень важную роль для мыслительного процесса представляют «специфические “навыки” мышления», возникающие в процессе целенаправленного решения определенных задач [37].

Таким образом, для того чтобы ученики могли выстраивать мысленные логические и алгоритмические цепочки, их нужно вовлекать в деятельность, связанную с анализом параллельного протекания внешних процессов, в том числе и программных.

В отличие от программирования последовательных вычислений, концептуальную основу которого составляет понятие алгоритма, реализуемого по шагам строго последовательно во времени, в параллельном программировании программа порождает совокупность параллельно протекающих процессов обработки информации, полностью независимых или связанных между собой статическими или динамическими пространственно-временными или причинно-следственными отношениями.[41]

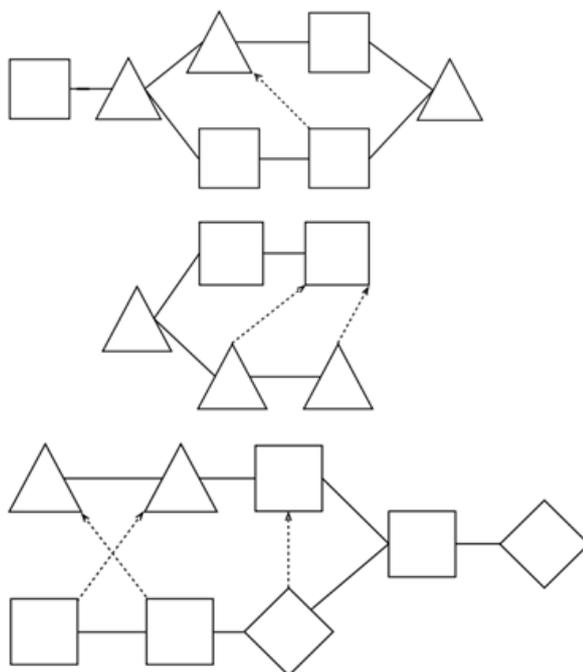
Параллельное программирование включает в себя все черты более традиционного, последовательного (структурного) программирования, но в

параллельном программировании имеются три дополнительных, чётко определённых этапа.

Определение параллелизма: анализ задачи с целью выделить подзадачи, которые могут выполняться одновременно.

Выявление параллелизма: изменение структуры задачи таким образом, чтобы можно было эффективно выполнять подзадачи. Для этого часто требуется найти зависимости между подзадачами и организовать их так, чтобы ими можно было эффективно управлять. [30].

В процессе изучения параллельной технологии программирования формируется параллельный стиль мышления, предполагающий способность к предварительному умозрительному «распараллеливанию» поставленной задачи — её анализу с целью выделения подзадач, «распараллеливанию» потоков данных - выделение потоков данных, которыми будут обмениваться выполняемые параллельно подзадачи. При этом в памяти формируются временные цепочки взаимосвязей понятийных и абстрактных объектов (рис. 4).



*Рис. 4. Формирование информационных связей понятийных и абстрактных объектов*

С учётом вышесказанного под параллельным стилем мышления будем подразумевать такой способ алгоритмической мыслительной деятельности, который при осознанной, специально организованной последовательности действий приводит к параллельному решению поставленной задачи.[41]

Структура параллельного стиля мышления может быть представлена упрощённой моделью (рис. 5).

В данной модели большую роль играют этапы, дополняющие привычный процесс последовательного решения. Органичное включение указанных этапов в процесс решения задачи, осознание их важности являются составляющими такого способа алгоритмической мыслительной деятельности, который назван параллельным стилем мышления.



*Рис. 5. Структура параллельного стиля мышления*

Помимо указанных в модели этапов: выделение подзадач, установление информационных зависимостей и организация информационного взаимодействия подзадач, очень важную роль играет понимание того, что все подзадачи на самом деле выполняются параллельно (одновременно).

Формирование понимания особенностей параллельных вычислений — это длительный и, как показывает опыт, сложный процесс, в ходе которого

«ломаются» привычные стереотипы как в трактовке (чтении), так и в составлении программного кода. При этом формируется новый способ алгоритмической деятельности — параллельный стиль мышления.

К предложенной модели необходимо сделать ещё несколько пояснений. Модель условно можно разделить на две части. Первая часть — это дополнительные (по отношению к структурному программированию) этапы параллельного решения задач, которые образуют довольно строгую последовательность. Вторая часть демонстрирует необходимость добиваться полного понимания процесса решения на каждом из этапов для любой задачи (эта необходимость отображена двунаправленными рёбрами, связывающими понимание с каждым из этапов, дополняющих обычный процесс решения).

Согласно предложенной модели структуры параллельного стиля мышления, можно выделить следующие этапы его формирования.

- a. Пропедевтический.
- b. Формирующий.
- c. Развивающий.

Сущность работы на пропедевтическом этапе видится в том, чтобы в учебных дисциплинах, а в частности на уроках информатики, применять систему дидактических игр для формирования представлений о параллельных процессах. Согласно этой системе учитель может и должен оказывать влияние, выражающееся в том, что акцент в деятельности учеников должен быть сделан на формировании тех мысленных действий, которые являются составными частями модели параллельного стиля мышления.

В результате учащийся, на пропедевтическом этапе развития параллельного стиля мышления должен овладеть следующие элементами:

- I. Умение видеть и выделять исполнителей в задаче:
  - a) понимание того, что задача решается совместно (Например: несколько строительных бригад строят дом);

b) в решаемых совместно заданиях умение определять количество исполнителей;

c) умение определять, исполнители являются однотипные или разнотипные (действия исполнителей являются равноправными или нет).

II. Планирование объема работы исполнителя:

a) умение определять, работы однотипные или разнотипные;

b) разделение задачи на части решаемые совместно или отдельно.

III. Разбиение подзадач на более мелкие в случае необходимости и обоснование таковой необходимости.

IV. Поиск ошибок в предложенном решении (разбиении) задачи.

V. Понимание нелинейного изменения скорости работы при увеличении количества исполнителей (Например: если 1 человек может приготовить суп за 50 мин, это не значит, что вдвоем его можно приготовить за 25 мин., а в пятером это займет всего 10 минут.)

VI. Умение находить оптимальное количество исполнителей для выполнения заданной работы.

VII. Умение найти оптимальную стоимость работы нескольких исполнителей.

VIII. Умение находить ресурсы в задаче и определять являются ли они совместными (Например: два ученика рисуют картину одними красками).

IX. Умение согласовывать деятельность исполнителей (по времени, результатам, ресурсам).

Рассмотрим некоторые примеры заданий, которые могут выступать как составные части фронтального (или группового) разбора задачи, или как самостоятельные тренировочные задания.

Для формирования умения видеть исполнители в задаче и выделять информационно-независимые подзадачи можно использовать индивидуальные или групповые задания, требующие разбить некоторую задачу на части (Приложение 3, 6, 8, 9, 10). Как продолжение работы может

выступать упражнением: изобразить таблицу или граф информационных зависимостей и проанализировать его с целью выделения групп подзадач, связанных между собой сильнее прочих или слабее. В этом случае мы способствуем формированию умения «видеть» информационные связи между подзадачами и в перспективе планировать организацию обменов данными между исполнителями и умение находить оптимальное количество исполнителей.

Одним из эффективных приёмов, направленных на понимание параллельной работы нескольких процессов, является пошаговое прохождение фрагмента параллельной задачи несколькими учениками, которые выступают в роли исполнителей. В ходе пошагового, одновременного выполнения работы нескольких процессов, ученики лучше понимают ход работы параллельной задачи и особенности взаимодействия между исполнителями (Приложение 1, 2, 3, 4, 8, 10, 12, 13).

Также с помощью дидактических игр можно демонстрировать механизмы согласования действий исполнителей (Приложение 7, 8, 9), конкуренцию исполнителей за ресурсы (Приложение 9, 11), взаимную блокировку исполнителей (Приложение 11). Эти свойства параллельных процессов плохо воспринимаются при простом словесном объяснении и нуждаются в наглядной демонстрации.

Применение систем дидактических игр при формировании представлений о параллельных процессах продолжает процесс формирования алгоритмического мышления в целом и параллельного стиля мышления как его составляющей в частности.

Сущность работы на формирующем этапе параллельного стиля мышления видится в том, чтобы на уроках информатики, применять методику параллельного изучения материала темы «Алгоритмизация и программирование». Согласно этой методике, теоретический материал даётся параллельно, т. е. одновременно излагается реализация одних и тех же

алгоритмических структур в разных языках программирования (Pascal, Visual Basic) и средах (ЛогоМиры, КуМир и т.д). Далее на практических занятиях ученики решают одну и ту же конкретную задачу средствами различных языков и сред.

Так же можно изучать последовательные и параллельные процессы одновременно, рассматривая одну и ту же задачу на разном количестве исполнителей, например в ЛогоМирах рассматривая программу с 2-мя черепашками.

Таким образом, можно сделать вывод о том, что в процессе формирования представлений о параллельных процессах изменяется стиль мышления школьников, который назван параллельным стилем. Указанный способ мышления поддается целенаправленному формированию с помощью деятельности ученика и преподавателя, спланированной согласно закономерностям развития мышления.

### **1.2.2. Понятие дидактической игры и ее роль в развитии школьников.**

Игра имеет большое значение в жизни ребенка, имеет то же значение, какое у взрослого деятельность, работа и служба. Игра только внешне кажется беззаботной и легкой. А на самом деле она властно требует, чтобы играющий отдал ей максимум своей энергии, ума, выдержки, самостоятельности.

Ученики повторяют в играх то, к чему относятся с полным вниманием, что им доступно наблюдать и что доступно их пониманию. Дидактическая игра, по мнению многих ученых, есть вид развивающей, социальной деятельности, форма освоения социального опыта, одна из сложных способностей человека. Советский педагог В.А. Сухомлинский подчеркивал, что «игра – это огромное светлое окно, через которое в духовный мир ребенка вливается живительный поток представлений, понятий об

окружающем мире. Игра – это искра, зажигающая огонек пытливости и любознательности».[43]

Н.К. Крупская во многих своих статьях упоминала о значении игры для познания мира, для нравственного воспитания детей. «Самодетельная подражательная игра, которая помогает осваивать полученные впечатления, имеет громадное значение, гораздо большее, чем что-либо другое». Ту же мысль высказывает А.М. Горький; «Игра – путь детей к познанию мира, в котором они живут и который призваны изменить».

Н.К. Крупская впервые в педагогике поставила вопрос о связи игры с трудом. Она доказала, что у детей нет резкой грани между этими видами деятельности; в игре, как и в работе, главное – постановка цели и ее выполнение. По мнению Н.К. Крупской, игра готовит детей к труду. Эту мысль развивает А.С. Макаренко. Он утверждает, что хорошая игра похожа на хорошую работу, их объединяет усилие мысли и рабочее усилие, радость творчества и чувство ответственности.[19]

«Чем же все-таки отличается игра от работы? Работа есть участие человека в общественном производстве, в создании материальных, культурных, иначе говоря, социальных ценностей. Игра не преследует таких целей, к общественным целям она не имеет прямого отношения, но к ним отношение косвенное, она приучает человека к тем физическим и психическим усилиям, которые необходимы для работы»[19]

В высказываниях великих педагогов – К.Д. Ушинского, Н.К. Крупской, А.С. Макаренко игра рассматривается как самостоятельная творческая деятельность детей.

«Очень важно не шаблонизировать игры, а давать простор детской инициативе. Важно, чтобы дети сами придумывали игры, ставили себе цели. Педагог не должен стеснять инициативу учеников, расхолаживать их, навязывать им те или иные игры». [36]

Многие исследователи пишут, что закономерности формирования умственных действий на материале школьного обучения обнаруживается в игровой деятельности детей. В ней своеобразными путями осуществляется формирование психических процессов: сенсорных процессов, абстракции и обобщения произвольного запоминания и т.д. Игровое обучение формирует способности учиться и развивает познавательную активность школьников.

Чем старше и более развиты дети, тем требовательнее они к предметам игры, тем больше сходства ищут с действительностью. Отсюда естественно возникает стремление самим сделать нужные вещи. Одна из тенденций развития игры – все большая связь ее с обучением. Задача учителя – поддержать это стремление ребенка к самостоятельному стремлению учиться и помочь ему в этом.

Игра в учебном процессе – это дидактический и социально важный «вид активности» (А.С. Макаренко).[19]

Собственно игровое взаимодействие учащихся создает условия мнимых искусственных ситуаций, вызывая соревнование и эмоциональное возбуждение, а вместе с тем и положительную мотивацию учебного познания, и концентрацию интеллектуальных усилий. Мобилизуя умственные способности учащихся, их воображение, внимание, память, дидактические игры «обычно связаны с возвышающим чувством удовольствия», и, следовательно, с условиями непроизвольного усвоения и запоминания учебного материала.

Игра – один из тех видов деятельности, в которой по средствам общения учащийся развивается как личность, у него формируются те стороны психики, от которых будут зависеть успешность его социальной практики, его отношения с окружающими людьми и с собой.

Использование дидактических игр и упражнений на уроках информатики способствует развитию познавательных интересов,

мыслительных процессов и положительной мотивации к обучению школьников.

Соединение в дидактической игре обучающей задачи с игровой формой, наличие готового содержания и правил даёт возможность педагогу более планомерно использовать дидактические игры как ценное средство воспитания умственной активности учеников.

В концепциях российских исследователей в последнее время отмечается тенденция к созданию и обоснованию серии развивающих дидактических игр, которые характеризуются гибкостью, инициативностью мыслительных процессов включенных в них субъектов, переносом сформированных у них умственных действий на новое содержание, отсутствием фиксированных правил и необходимостью выбора способов решения задачи. По мнению А.В. Менджерицкой, сущность таких дидактических игр заключается в том, что «дети решают умственные задачи, предложенные им в занимательной форме,... преодолевая при этом определенные трудности», что повышает их интеллектуальную активность. Также в отечественной психолого-педагогической науке и практике дидактическая игра рассматривается как средство закрепления знаний, полученных на занятиях, как одна из форм обучения, как средство образовательной работы (А.И. Сорокина). Данный подход определяется задачами обучения школьников, предполагающими не только формирование у учеников определенного объема знаний, но и научение использованию полученных сведений, развитие активности, самостоятельности мышления и иных общих познавательных способностей.

В процессе формирования у школьников представлений о параллельных вычислениях дидактические игры выполняют особую роль, обусловленную преимущественно практической деятельностью учащихся и сравнительно небольшим объемом учебного времени, отводимого на изучение сложных теоретических вопросов. Использование дидактических

игр на уроках информатики при изучении параллельных процессов позволит учителю более выразительно и доходчиво донести до школьников наиболее трудные и сложно усваиваемые вопросы программного материала, активизировать их наблюдения, способствовать более глубокому восприятию и запоминанию информации; расширит кругозор и повысит интерес школьников к данному направлению и к самому предмету.

Отсюда следует, что использование дидактических игр и упражнений на уроках информатики при изучении основ параллельных вычислений является неотъемлемой частью учебного процесса. Безусловно, игры – это весьма активная деятельность, поэтому деятельностный подход при разработке системы дидактических игр, нацеленных на формирование первичных представлений о параллельных процессах, представляется наиболее оптимальным.

### **1.2.3. Формирование представлений о параллельных процессах в рамках деятельностного подхода.**

Современная литература не дает единого подхода к формированию определения понятия деятельность.

Развернутое определение этого понятия в философии сформулировано Э.Г. Юдиным: «Деятельность есть специфически человеческая форма активного отношения к окружающему миру, содержание которой составляет целесообразное изменение и преобразование этого мира на основе определённой потребности, мотива». [51, стр.56]

Эта позиция стала отправной точкой в основе концепции взаимодействия субъектов образовательного процесса. Исследования Э.Г. Юдина позволяют установить атрибуты деятельности: целеполагание, предметность, осмысленность, преобразующий характер.

По мнению психолога Леонтьева А.Н. с полным основанием о деятельности можно говорить только тогда, когда «это совокупность активных действий, которые вызываются одним мотивом». [26, стр.257]

В педагогике «Деятельность - это внутренняя (психическая) и внешняя (физическая) активность человека, регулируемая сознаваемой целью».

Обратив внимания на предложенные определения можно увидеть общее: деятельность предполагает активные мотивированные действия.

Под термином «деятельностный подход» следует понимать: «планирование и организацию учебного процесса, в котором главное место отводится активной и разносторонней, в максимальной степени самостоятельной познавательной деятельности школьников» [1 стр. 26] .

Деятельностный подход в последние годы интенсивно разрабатывался в философии, психологии, педагогике (Л.С. Выгодский, И. С. Якиманской, П.Я.Гальперин, В.В.Давыдов, А.Н.Леонтьев и др.). Этот подход лежит в сфере интересов учителей различных предметов.

По П. Я. Гальперину «Действие – это единица анализа деятельности учащихся» [10, стр.12]. Оно представляет собой целостную систему взаимосвязанных элементов. В ходе выполнения действия эти элементы обеспечивают три основные функции: ориентировочную, исполнительную, контрольную.

Ориентировочная часть является центральной, намечает пути и способы действия, исполнительная часть их реализует, а на контрольно-корректировочной части происходит оценка результата и ликвидация пробелов в знаниях.

В своей работе П. Я. Гальперин выделяет три типа учения:

а) по первому типу учения объяснение любого действия сводится к его однократной демонстрации, показу образца и к неполному словесному описанию по ходу показа. Дефицит начальных условий становится причиной большого числа проб и ошибок. При таком типе обучения у большинства

детей происходит накопление узких знаний и умений и не развивает мышление;

б) обучение по второму типу характеризуется тем, что всю систему необходимых условий заранее излагают в готовом виде учащимся и активность ученика на усвоение готового знания; воспитание познавательного интереса к знанию не происходит;

с) третий тип учения характеризуется полной системой условий, которая не задается в готовом виде, строится самим учащимся, который исходит из проблем, возникающих при столкновении фактов. В этом типе учения развитие мышления проявляется в самостоятельном распространении усвоенных приемов исследования на дальнейшие разделы того же учебного предмета и за его границы.

Строя обучения основам параллельных процессов на основе деятельностного подхода необходимо руководствоваться следующими положениями:

а) обучение должно быть построено по третьему типу;

б) в деятельности учащихся преобладают активные действия на ориентировочном и исполнительском этапах;

с) деятельность учащихся организована таким образом, что созданы необходимые условия для формирования и дальнейшего развития мыслительных операций.

### **§3. Понятийные проблемы линии алгоритмизации в школьном курсе информатики.**

В соответствии с общей структурой школьного образования сегодня выстраивается многоуровневая структура предмета «Информатика и ИКТ». Начальный курс (2-4-й классы) во многих школах изучается интегративно в рамках других учебных предметов. Основной курс (5-9-й классы) и профильный курс (10-11-й классы) изучаются как отдельный предмет.

Профильный курс изучения основ программирования предполагает развитие объектного стиля мышления на базе изучения объектно-ориентированных языков программирования. Задача основного курса - изучение основ алгоритмизации и программирования, являющихся подготовительным этапом к профильному курсу. На этом этапе возможно развитие алгоритмического, логического мышления учеников, а также формирование операционного типа мышления.

В связи со спецификой этапа основного образования как самого продолжительного в структуре основного курса информатики выделяются две последовательные части: вводная (5-6-й классы), одной из целей которой является пропедевтика понятий базового курса информатики, и базовая (7-9-й классы) [3].

К настоящему времени в школьной информатике имеются значительные учебно-методические наработки для разных возрастных групп учащихся, изданы учебники и учебные пособия.

Методика обучения основам алгоритмизации и программирования представлена в рамках курсов широко известных авторских коллективов:

1. УМК «Алгоритмика», авторы Звонкин А.К., Ландо С.К. и др., 5-7-й классы. [24]
2. УМК по курсу Информатика и ИКТ, автор Босова Л.Л., 5-й, 6-й, 7-й классы. [5,2]

3. УМК «Информатика и ИКТ. Начальный уровень», авторский коллектив под редакцией профессора Макаровой Н.В., 5-й, 6-й классы.

4. УМК «Информатика и ИКТ. Базовый уровень», авторский коллектив под редакцией профессора Макаровой Н.В., 7-9-й классы. [28]

5. УМК «Информатика. Базовый курс», авторы Семакин И.Г., Залогова Л.А. и др., 7-9-й классы. [38]

6. «Информатика. Базовый курс», автор Угринович Н.Д., 7-й, 8-й, 9-й классы. [46]

Рассмотрим подробнее курсы этих авторов.

**1. «Алгоритмика», авторы Звонкин А.К., Ландо С.К. и др., 5-7-й классы. [24]**

Основной целью данного курса является формирование у школьника основ алгоритмического мышления. Под алгоритмическим мышлением понимается умение решать задачи различного происхождения, требующие составления плана действий для достижения желаемого результата. Курс рассчитан на обучение в течение 1 года для безкомпьютерного варианта обучения. В комплект УМК входит интерактивный задачник «Алгоритмика 2.0». При проведении курса с использованием компьютера целесообразно увеличить время обучения в 1,5 раза.

Обучение школьников основам алгоритмического мышления базируется на понятии Исполнитель. Главные действующие лица программы – исполнители: Робот, Черепаха, Водолей, Кузнечик и другие. Каждый из них понимает небольшое количество простых команд, при помощи которых ими можно управлять. В каждой конкретной задаче требуется заставить исполнителя совершить те или иные действия. Например, Роботу требуется пройти лабиринт, обходя препятствия и закрашивая по пути, отмеченные заранее клетки, Черепахе – нарисовать сложную геометрическую фигуру, Водолею – отмерить нужное количество воды, имея только емкости определенного размера. Решение задачи, как правило, заключается в

составлении алгоритма для исполнителя. Исполнители курса традиционны. Исключение составляет Исполнитель Директор строительства. Это одна из первых попыток познакомить школьника с понятием параллельного программирования.

Схема подачи материала в курсе следующая: от примера к понятию, от частного к общему. В курсе осваиваются следующие понятия: Исполнитель, среда Исполнителя, система команд, алгоритм, цикл; также рассматриваются задачи всех алгоритмических конструкций без использования переменных.

## **2. УМК по курсу Информатика и ИКТ, автор Босова Л.Л., 5-й, 6-й, 7-й классы. [5,2]**

В данном курсе изучение темы «Алгоритмизация и программирование» планируется в 6-м, 7-м классах.

В 6-м классе выделено 9 часов на изучение понятий исполнитель, система команд исполнителя (СКИ), алгоритм (линейный алгоритм, алгоритм ветвления и циклический алгоритм) в безкомпьютерном варианте с использованием заданий в рабочей тетради. Также в курсе рассматриваются задачи построения различных геометрических фигур с помощью графических исполнителей DRAW, CIRCLE, LINE в среде программирования QBasic.

В 7-м классе выделено 7 часов на изучение тем алгоритмики. Рассматриваются традиционные исполнители Чертежник и Робот. Вводится понятие вспомогательных алгоритмов, рассматриваются алгоритмические конструкции цикл повторить N раз, цикл «пока», ветвление.

Данный курс способствует развитию алгоритмического мышления и позволяет подготовить учащихся к дальнейшему изучению среды программирования QBasic.

### **3. УМК «Информатика и ИКТ. Начальный уровень», авторский коллектив под редакцией Макаровой Н.В., 5-й, 6-й классы.**

В 5-м классе данного курса рассматриваются понятия алгоритм (линейный алгоритм, алгоритм ветвления и циклический алгоритм). Все данные понятия изучаются на примерах построения графических объектов в среде Paint. В каждой конкретной задаче разрабатывается алгоритм и в соответствии с ним создается графический объект. На изучение данной темы отводится 10 часов.

В 6-м классе изучается курс программирование и моделирование в среде ЛогоМиры. Целью данного курса является развитие творческого потенциала учащихся, развитие логического и алгоритмического мышления. Ученики осваивают основы программирования, выполняя сюжетные задания.

Учащиеся знакомятся с понятием команды и входных параметров, понятием программы и организацией конечного цикла в среде ЛогоМиры, исполнителем среды Черепашкой, основными объектами среды: бегунками, кнопками и тд., датчиками, определяющими состояние Черепашки, датчиком случайных чисел. Среда ЛогоМиры дает возможность моделировать движение Черепашки, создавать анимационные проекты. Мультимедийные возможности ЛогоМиров позволяют создавать проекты с мультипликацией, видеофрагментами и звуковым сопровождением. На изучение данной темы отводится 24 часа.

Курс развивает алгоритмическое и логическое мышление, умение составлять алгоритмы и позволяет увидеть их реализацию, способствует повышению творческого потенциала учащихся.

### **4. УМК «Информатика и ИКТ. Базовый уровень», авторский коллектив под редакцией Макаровой Н.В., 7-9-й классы. [28]**

Данный курс базового уровня является продолжением начального уровня 5-го и 6-го классов. В зависимости от выделенных часов на курс Информатика и ИКТ возможны различные варианты изучения тем

алгоритмизации и программирования. При 2-х часовом курсе, на протяжении 3 лет предполагается изучение темы «Основы алгоритмизации» на базе языков Pascal или Visual Basic в объеме 17 часов и темы «Среда программирования» ЛогоМиры в объеме 14 часов.

В теме «Основы алгоритмизации» рассматриваются следующие понятия: алгоритм (линейный алгоритм, алгоритм ветвления и циклический алгоритм) свойства алгоритмов, вспомогательный алгоритм, назначение процедуры, представление алгоритма в виде блок-схемы, рассматриваются стадии создания алгоритма.

Изучая тему «Программирование» в среде ЛогоМиры, ученики знакомятся с:

- инструментарием данной среды;
- программами для реализации типовых конструкций алгоритмов (последовательного, циклического, разветвляющегося);
- понятиями процедуры и модуля, процедуры с параметрами;
- функциями;
- инструментами логики при разработке программ.

Методика разработки простых программ в среде ЛогоМиры позволяет развить у учащихся навыки решения задач с применением алгоритмического, системного и объектно-ориентированного подходов к решению задач; формирует алгоритмическое и логическое мышление; способствует развитию интереса школьников к обучению и повышению их творческого потенциала.

#### **5. УМК «Информатика. Базовый курс», авторы Семакин И.Г., Залогова Л.А. и др., 7-9-й классы. [38]**

Спецификой данного курса является то, что он построен по двухуровневому принципу: 1 уровень - материал, соответствующий минимальному содержанию базового курса, 2 уровень - дополнительный материал, расширяющий содержание разделов первого уровня, используется при изучении курса по углубленному варианту. Этот принцип построения

курса характерен и для темы «алгоритмизация и программирование». Соответственно, планирование данной темы соответствует 16 и 42 часам изучения.

В основе базового варианта изучения основные понятия: алгоритм (линейный алгоритм, алгоритм ветвления и циклический алгоритм), свойства алгоритма, исполнитель и система команд исполнителя (СКИ) рассматриваются с использованием алгоритмического языка – (АЯ). Также изучается язык блок-схем, вспомогательные алгоритмы, метод пошаговой детализации. Кратко изучаются язык программирования высокого уровня Pascal.

Углубленный уровень предполагает дополнительное изучение темы «Логическое программирование» на языке Prolog, а также программирование на языке программирования высокого уровня Pascal.

В курсе предполагается решение большого количества задач, позволяющих усвоить учащимися основы алгоритмизации и программирования на высоком уровне. Задачник-практикум дает множество материала для организации практической работы на уроках и домашней работы учеников. Большое число различных заданий предоставляет учителю возможность варьировать содержание курса по времени и уровню сложности.

При изучении данного курса учащиеся смогут:

- выполнять трассировку заданных простых алгоритмов;
- строить блок-схемы несложных алгоритмов;
- использовать школьный алгоритмический язык для описания алгоритмов;
- работать с готовой программой на одном из языков программирования высокого уровня (Pascal);
- составлять несложные программы решения вычислительных задач;
- осуществлять отладку и тестирование программы.

## **6. «Информатика. Базовый курс», автор Угринович Н.Д., 8-й, 9-й классы. [46]**

В данном курсе в 9-м классе предполагается изучение темы «Основы алгоритмизации и объектно-ориентированного программирования» в среде Visual Basic на ее изучение отводится 14 часов. Учащиеся изучают объекты среды, свойства, методы, события; рассматривают событийные и общие процедуры, операторы ветвления, выбора, цикла; знакомятся с понятиями переменная, арифметические, строковые и логические выражения.

При изучении данного курса учащиеся смогут объяснить:

- структуру основных алгоритмических конструкций и использовать их для построения алгоритмов;
- определить основные типы данных и операторы;
- разработать и записать на языке программирования типовые алгоритмы;
- создавать проекты с использованием визуального объектно-ориентированного программирования.

Объектно-ориентированный подход к решению задач позволяет развить у учащихся алгоритмическое мышление и сформировать объектный стиль мышления, что в свою очередь способствует подготовке учащихся к дальнейшему изучению среды программирования Visual Basic.

Проанализировав содержание УМК, можно сделать вывод, что все они реализуют требования ФГОС по количеству изученных терминов, и уровню изложения материала.

Из всех проанализированных УМК нигде не упоминается о параллельных вычислениях и суперкомпьютерных технологиях, за исключением УМК «Алгоритмика», авторы Звонкин А.К., Ландо С.К. и др., 5-7-й классы [24], в котором рассматривается составление программы для исполнителя «Директор строительства». Это одна из первых попыток познакомить школьника с понятием параллельного программирования.

Основная задача изучения раздела «Алгоритмы и системы программирования» – это не столько изучение алгоритмов и систем программирования, сколько развитие алгоритмического стиля мышления, что дает нам поле и для развития параллельного стиля мышления.

В изучении раздела «Алгоритмы и системы программирования» необходимо заняться более подробным и углубленным изучением области суперкомпьютерных технологий.

Сделать это можно только одним путем — включением соответствующих параграфов и элементов пропедевтики основ параллельных процессов в раздел «Алгоритмы и системы программирования». Значит, нужна соответствующая методика начального знакомства с параллельным программированием на уровне младшей и средней школы.

Современный этап развития computer science связан с массовым распространением параллелизма вычислений на всех уровнях (многомашинные кластеры, многопроцессорные ЭВМ, многоядерные процессоры).

Массовое распространение параллелизма влечет серьезные последствия, которые еще предстоит выявить и проанализировать. Начнем с перечисления некоторых теоретических проблем.

Современная теория алгоритмов создавалась в расчете на понятие последовательного алгоритма. Каким образом отразится на понятии алгоритма отказ от требования последовательности выполнения шагов?

По крайней мере последние 20 лет понятие «алгоритм» вводилось в школе в неразрывной связке с понятием «исполнитель». Для последовательного алгоритма это естественно. Как быть с алгоритмом параллельным? Его выполняет один исполнитель или группа исполнителей? Для конкретности в качестве примера рассмотрим обучающую компьютерную программу «Танковый экипаж» предложенную

Плаксиным М.А., [23] который так же выделяет понятийные проблемы линии алгоритмизации в школьном курсе информатики. В этой программе от учащегося требуется запрограммировать действия экипажа танка, состоящего из трех человек: наводчика, водителя и заряжающего. Каждый из них имеет свою систему команд. Для того, чтобы выполнить боевую задачу (поразить все цели), все члены экипажа должны действовать согласованно. Подробное описание игры «Танковый экипаж» представлено в приложении 12.

Пример игрового поля программы «Танковый экипаж» см. на рис. 6.

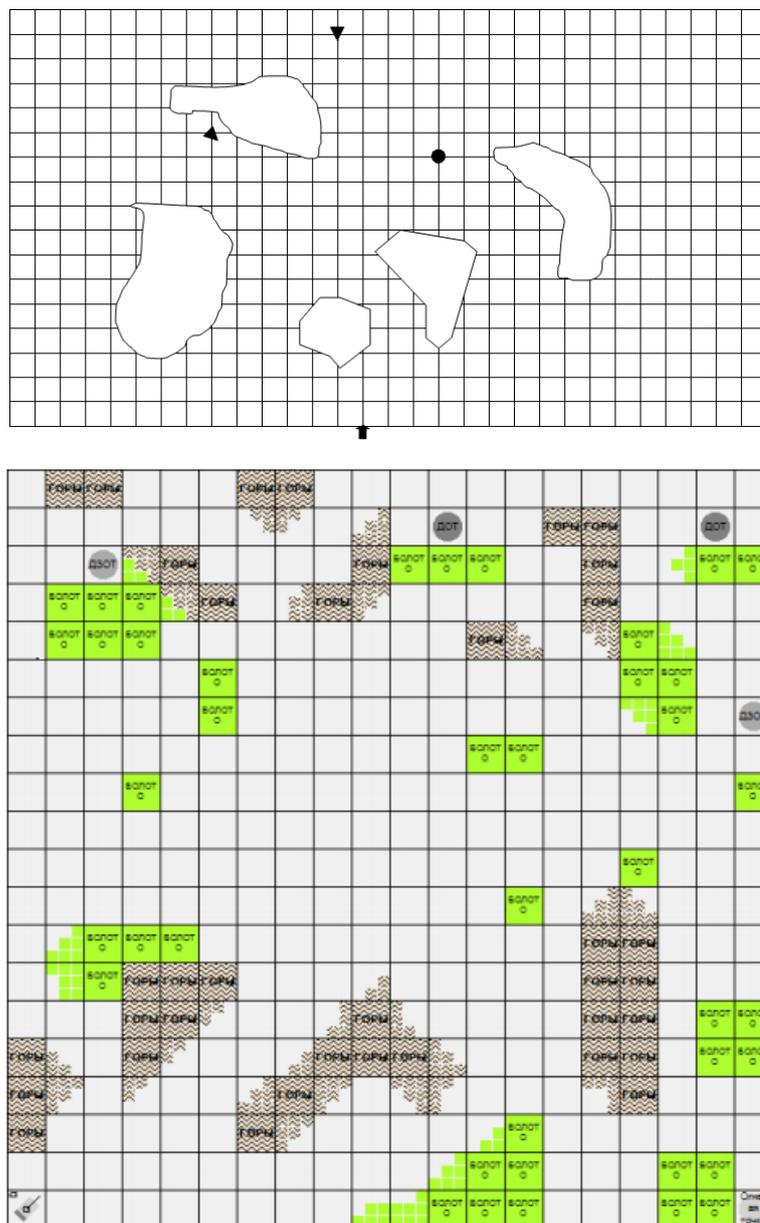


Рис. 6. Пример поля боя для игры «Танковый экипаж»

Вопрос: надо ли рассматривать этих трех действующих лиц как независимых исполнителей или как три составные части (устройства) одного сложного исполнителя? Для экипажа танка более естественным представляется второй вариант, поскольку ни один персонаж сам по себе выполнить задание не в состоянии. Но как быть, если игра будет усложнена, и боевая задача будет поставлена сразу для двух танков? Для трех танков? Трех членов одного экипажа вполне можно рассматривать как три части одного исполнителя. Но каждый экипаж, очевидно, является самостоятельным исполнителем. Значит, параллельный алгоритм для нескольких танков будет выполняться сразу группой исполнителей. Получается, что для параллельного алгоритма рассматривать надо обе возможности: выполнение параллельных действий одним исполнителем и группой исполнителей. В случае танкового экипажа границу провести просто. Исполнитель — это тот, кто в состоянии решить поставленную задачу. Этот исполнитель может состоять из нескольких компонент, каждая из которых выполняет некую часть задания, но не может самостоятельно без помощи других компонент выполнить задание целиком. Но всегда ли разделение «целых исполнителей» и частей сложного исполнителя будет также просто — сейчас сказать нельзя.

Выделение частей исполнителя, способных к самостоятельным действиям, требует как-то эти части назвать. Причем название должно допускать рекурсию, поскольку действующие части исполнителя сами могут иметь сложную структуру.

Нужно договориться о термине для обозначения группы совместно действующих исполнителей. Термин «команда» не годится, ассоциируется с «системой команд исполнителя» и с «командами центрального процессора». «Коллектив исполнителей»? «Бригада исполнителей»?

Понятие команды удобно расширить до «линейки команд». Однако понятие «линейки команд» хорошо работает только для линейных

алгоритмов. В остальных случаях линейки формируются динамически. Изобразить их в виде наглядной таблицы невозможно.

Среди свойств алгоритмов выделяется новая практически значимая характеристика: способность к распараллеливанию. Уточняющий вопрос — о возможной степени распараллеливания (до какой степени имеет смысл увеличивать количество процессоров при выполнении данного алгоритма).

Отдельный вопрос — методы распараллеливания уже существующих последовательных алгоритмов.

## Выводы по главе 1

В результате изучения и анализа методической, психологической и педагогической литературы, а также литературы по параллельным вычислениям можно сделать следующие выводы.

1. Требования современного информационного общества, определяют необходимость совершенствования предметной подготовки школьников в области современных суперкомпьютерных технологий, через ввод темы «параллельные вычисления» в школьный курс информатики и развитие особого – параллельного стиля мышления. Введение указанной темы в школьный курс информатики сформирует первичные представления о параллельных процессах, а также может способствовать развитию абстрактного, параллельного и алгоритмического мышления.

2. Деятельностный подход имеет важное значение для решения задачи повышения качества изучения абстрактного материала. Он направлен на организацию активно-деятельностного обучения с ориентацией на развитие личности учащихся, их умственных и творческих способностей, создание условий для саморазвития учащихся, обеспечения базы для «запуска механизмов» самообучения.

3. Для того чтобы подготовить специалистов в области параллельного программирования, необходимо привить навыки решения подобных задач на интуитивном уровне. Нужно проводить пропедевтику знаний по данному направлению, т.е. на интуитивном уровне вводить элементы взаимодействия параллельных процессов на примере их собственной деятельности. Такая пропедевтика может послужить базой для изучения параллельных процессов в вычислительных системах в дальнейшем, а также может служить одной из форм организации коллективной работы по принципу функционирования кластера.

4. Применение игровых методов на уроках информатики при формировании у школьников представлений о параллельных вычислениях

обусловлено широкими образовательными возможностями игры. Использование развивающих дидактических игр позволяет учителю более выразительно и доходчиво донести до школьников наиболее трудные и сложно усваиваемые вопросы программного материала, активизировать наблюдения учеников, способствовать более глубокому восприятию и запоминанию информации, развитию познавательных интересов, мыслительных процессов и положительной мотивации к обучению. В связи с этим применение различных видов игр, игровых приемов и игровых форм в процессе пропедевтической подготовки школьников основам параллельных процессов является весьма целесообразным.

выявлены понятийные проблемы линии алгоритмизации в школьном курсе информатики.

5. Выявлены понятийные проблемы линии алгоритмизации в школьном курсе информатики: что считать параллельным алгоритмом? До какой степени имеет смысл увеличивать количество исполнителей при выполнении параллельного алгоритма? Требуется доработки традиционное понятие системы команд исполнителя (СКИ) и само понятие команды.

## **ГЛАВА 2. Система дидактических игр, как средство формирования первичных представлений учащихся о параллельных процессах.**

### **§1. Дидактические принципы, цель и содержание системы игр.**

Система дидактических игр обучения школьников основам параллельных процессов построена на основных и значимых дидактических принципах обучения.

Как отмечает М.А. Данилов «дидактические принципы обучения - это основные положения, определяющие содержание, организационные формы и методы учебного процесса в соответствии с его общими целями и закономерностями» [31].

В качестве базовых дидактических принципов, нами приняты:

- Принцип *наглядности*, обусловленный включением в педагогический процесс наглядных средств обучения, использование которых способствует повышению интереса учащихся к знаниям и делает процесс обучения более легким. Многие сложные элементы параллельных процессов при умелом использовании наглядности становятся доступными и понятными для учащихся. По мнению Ушинского, наглядное обучение повышает внимание учащихся, способствует более глубокому усвоению знаний [31].

- Принцип *доступности*, соответствие которому обязывает определённым образом адаптировать сложную для восприятия и понимания информацию о понятиях параллельных вычислений.

- Принцип *систематичности и системности* - заключается в обеспечении последовательного усвоения учащимися определенной системы знаний, систематическое прохождение обучения. Принцип требует постоянного повторения изученного материала. По мнению Коменского, в

природе, как и в обучении все должно быть взаимосвязанным и целесообразным [31].

- Принцип *научности* - реализуется в анализе учебного материала, выделении в нем важных идей, использовании достоверных научных знаний, фактов и примеров, а также стандартных научных терминов. Принцип научности требует, чтобы содержание образования, реализуемое как в учебное, так и во внеучебное время, было направлено на ознакомление обучаемых с объективными научными фактами, явлениями, законами, основными теориями и концепциями той или иной отрасли, приближаясь к раскрытию ее современных достижений и перспектив развития.

#### **Принцип наглядности.**

Значимость принципа наглядности подчеркивал еще сам Я.А.Коменский. Он писал «... это «золотое правило» для учителей. Чем больше знание опирается на ощущение, тем более оно достоверно. Наглядность дает возможность сделать школу жизненной, учит учащихся понимать и изучать действительность самостоятельно.» [22]. Со временем значимость его слов не только не уменьшилось, а увеличивалось от эпохи к эпохе, так К.Д. Ушинский писал «...Педагог, желающий что-нибудь прочно запечатлеть в детские памяти, должен заботиться о том, чтобы как можно больше органов чувств глаза, ухо, голос, чувство мускульных движений и даже, если это возможно, обоняние и вкус, приняли участие в акте запоминания... Чем больше органов наших чувств принимает участие в восприятии какого-либо впечатления, или группы впечатлений, тем прочнее ложатся эти впечатления в нашу механическую, нервную память, вернее сохраняются ею и легче потом вспоминаются» [22].

Принцип наглядности основан на том, что в процессе познания окружающей действительности участвуют все органы чувств человека. Поэтому принцип наглядности выражает необходимость формирования у учащихся представлений и понятий на основе всех чувственных восприятий

предметов и явлений. Однако пропускная способность у органов чувств или «каналов связи» человека с окружающим миром различна. По мнению некоторых специалистов, если, например, орган слуха пропускает 1000 условных единиц информации за единицу времени, то орган осязания за ту же единицу времени пропускает 10 000 условных единиц информации, а орган зрения – 100 000, т.е. около 80 % сведений об окружающем мире человек получает с помощью зрения.

Таким образом, отмечая наибольшую пропускную способность информации у органов зрения, мы ставим принцип наглядности на первое место. Однако он предусматривает не только опору на зрение, но и на все другие органы чувств. На это положение обращал внимание и великий русский педагог К.Д. Ушинский. Он утверждал: «...чем большее количество органов чувств принимает участие в восприятии какого-нибудь впечатления, тем прочнее оно закрепляется в нашей памяти...». Физиологи и психологи объясняют это положение тем, что все органы чувств человека взаимосвязаны. Экспериментально доказано, что если человек получает информацию одновременно с помощью зрения и слуха, то она воспринимается более обостренно по сравнению с той информацией, которая поступает только через посредство зрения, или только через посредство слуха.[11]

Применение наглядных и технических средств обучения способствует не только эффективному усвоению соответствующей информации, но и активизирует познавательную деятельность обучающихся; развивает у них способность связывать теорию с практикой; повышает интерес к учению и делает его более доступным.

Наглядные пособия сами по себе в процессе обучения никакой особой роли не играют, они эффективны только в сочетании со словом учителя. Очень часто принцип наглядности воспринимается педагогами как необходимость непосредственных наблюдений учащимися за

определенными явлениями. Однако не всякое восприятие и не всегда продуктивно, оно может быть таковым только при активном мышлении, при возникновении вопросов и стремлении учащихся найти на них ответы. Еще Н. Пирогов в свое время отмечал, что «ни наглядность, ни слово сами по себе, без умения с ними обращаться как надо... ничего путного не сделают» [33].

Существуют разные способы сочетания слова и наглядности, которые подробно проанализированы и обобщены Л.В. Занковым в его книге «Наглядность и активизация учащихся в обучении» [15]. Наиболее типичными из них являются:

- при помощи слова учитель сообщает сведения об объектах и явлениях и потом, демонстрируя соответствующие наглядные пособия, подтверждает правдивость своей информации;
- при помощи слова учитель руководит наблюдениями учащихся, а знания о соответствующих явлениях они приобретают в процессе непосредственного наблюдения за этим явлением.

Очевидно, что второй способ более эффективен, нежели первый, так как он ориентируется на активизацию деятельности учащихся, но чаще всего используется именно первый. Это объясняется тем, что первый способ более экономичен по времени, он проще для учителя и требует меньше времени при подготовке к занятиям.

С одной стороны, наглядность может быть использована с целью обогащения чувственного опыта учащихся. С другой стороны, наглядность может использоваться лишь для разъяснения сущности явления. Когда младших школьников мы обучаем счету, то не нужны плакаты с красивыми корабликами или самолетами, здесь необходимы плакаты с обычными карандашами, так как в противном случае мы привлечем внимание детей не к количеству предметов, не к счету, а к самолетикам, к самой картинке. Таким

образом, обучение с использованием средств наглядности требует от учителя рациональной организации уроков.

### **Принцип доступности.**

Принцип доступности обучения вытекает из требований, выработанных многовековой практикой обучения, с одной стороны, закономерностей возрастного развития учащихся, организации и осуществления дидактического процесса в соответствии с уровнем развития учащихся, с другой.

В основе принципа доступности лежит закон тезауруса: доступным для человека является только то, что соответствует его тезаурусу. Латинское слово *tesaurus* в переводе означает «сокровище». В переносном значении под этим понимается объем накопленных человеком знаний, умений и способов мышления. [35]

Также можно выделить и другие закономерности, лежащие в основе принципа доступности:

1. Доступность обучения определяется возрастными особенностями и зависит от их индивидуальных особенностей;
2. Доступность обучения зависит от организации учебного процесса, применяемых методов обучения и связана с условиями протекания процесса обучения;
3. Чем выше уровень умственного развития обучающихся и имеющийся у них запас представлений и понятий, тем успешнее они могут продвинуться вперед при изучении новых знаний;
4. Постепенное нарастание трудностей обучения и приучение к их преодолению положительно влияют на развитие обучающихся;
5. Обучение на оптимальном уровне трудности положительно влияет на темпы и эффективность обучения, качество знаний.

Известны классические правила, относящиеся к практической реализации принципа доступности, сформулированные еще Я.А. Коменским:

от легкого к трудному, от известного к неизвестному, от простого к сложному. [35]

Принцип доступности требует, чтобы объем и содержание учебного материала были по силам учащимся, соответствовали уровню их умственного развития и имеющемуся запасу знаний, умений и навыков.

Доступность не следует понимать как учение без трудностей. Она не исключает приучение учащихся к преодолению трудностей в учебной деятельности. Это понятно, так как учебная работа требует определенных усилий учащихся в достижении поставленных целей. Суть вопроса заключается не в том, чтобы обходить трудности, а в том, чтобы эти трудности не подрывали, а развивали силы ученика и способствовали повышению результатов учебных занятий.

Реализация принципа доступности предполагает выполнение следующих условий - дидактических правил:

1. Следовать в обучении от простого к сложному;
2. От легкого к трудному;
3. От известного к неизвестному.

Отсюда следует, что строгое соблюдение в обучении принципа систематичности и последовательности предопределяет успешную реализацию принципа доступности. [35]

Следовать в обучении от простого к сложному означает, что изучение учащимися фактов, явлений, закономерностей, понятий и т. п. должно начинаться с наиболее простых, с тем чтобы подготовить их к пониманию более сложных. Это положение касается как теоретического, так и практического учебного материала.

Принцип доступности в обучении привлекает к себе особое внимание также в связи с проблемой индивидуального подхода к учащимся и условиях массового обучения.

Принцип доступности требует, чтобы обучение строилось на основе учета возрастных возможностей учащихся. С его помощью регулируется уровень сложности учебного материала, определяется выбор методических подходов изложения его на уроке, правильная дозировка домашних заданий. Слишком упрощенное содержание обучения снижает его развивающие и воспитательные возможности. Поэтому рекомендуется (по Л. В. Занкову) [16], чтобы содержание заданий для учащихся находилось в "зоне их ближайшего развития".

### **Принцип систематичности и системности.**

Впервые этот принцип, как и многие другие, был применен Я.А. Коменским, считавшим, что, как и в природе, в обучении все должно быть взаимосвязанным и целесообразным.

Эти идеи были позднее развиты Песталоцци и К.Д. Ушинским, который отмечал, что «голова, наполненная бессвязными знаниями, похожа на кладовую, где все в беспорядке и где сам хозяин ничего не отыщет».[34]

Развитие системного подхода к обучению позволило более четко структурировать учебный материал, создать комплекты учебных и наглядных пособий по изучаемым учебным предметам. Системное структурирование требует вычленения в изучаемом материале ведущих понятий и категорий, установления их связей с другими понятиями и категориями.

Процесс обучения должен проводиться строго последовательно, с соблюдением правила идти "от незнания к знанию, от неумения к умению". Об этом образно писал Я. А. Коменский: "Природа не делает скачков, а идет вперед постепенно... Так подвигается вперед и тот, кто строит дом. Он начинает не с крыши и не со стен, а с фундамента. А заложив фундамент, не покрывает его крышей, а воздвигает стены. Словом, как в природе все сцепляется одно с другим, так и в обучении нужно связывать все одно с другим именно так, а не иначе..." [34].

Последовательность в обучении обеспечивает доступность учебного материала, прочность его усвоения, постепенное нарастание трудностей и развитие познавательных возможностей обучаемых. Она реализуется:

- в психологически и педагогически выверенном распределении учебного материала по каждой учебной дисциплине;
- в прохождении тем учебного материала в определенном порядке;
- в обоснованных действиях педагогов по развитию различных личностных качеств.

В образовательной практике принцип преемственности, последовательности и систематичности реализуется в процессе тематического планирования, когда педагог намечает последовательность изучения отдельных разделов, тем, вопросов, отбирает содержание, намечает систему уроков и других форм организации процесса обучения, планирует усвоение, повторение, закрепление и формы контроля. При поурочном планировании учитель располагает содержанием темы таким образом, чтобы исходные понятия изучались ранее, а тренировочные упражнения следовали бы за изучением теории.

### **Принцип научности.**

Принцип научности, в отличие от многих других, долгое время не был необходимым в организации обучения, так как и сама наука не играла существенной роли в процессе трудовой деятельности человека. Поэтому он не выделялся Я.А. Коменским. [11]

В дальнейшем, с повышением роли знаний в практической деятельности человечества, в общественном производстве наряду с обыденными житейскими толкованиями явлений окружающего мира стало необходимым его научное понимание [11]. Вслед за этим встал вопрос о введении принципа научности и в систему образования.

Главной целью принципа научности является то, чтобы учащиеся понимали, что все в этом мире подчинено законам и что знание их необходимо каждому живущему в современном обществе. [11]

Принцип научности обучения предполагает соответствие содержания образования уровню развития современной науки и техники.

Суть принципа научности состоит в том, что содержание образования должно быть научным и иметь мировоззренческую направленность. Для его реализации учителю необходимо: глубоко и доказательно раскрывать каждое научное положение изучаемого материала, не допуская ошибок, неточностей и механического зазубривания учащимися теоретических выводов и обобщений; показывать значение изучаемого материала для понимания современных общественно-политических событий и их соответствия интересам и стремлениям народа. [31]

Правила принципа научности: [11]

1. Принцип научности требует, чтобы предлагаемый учебный материал отвечал современным достижениям науки. Упрощение сложных научных положений не должно приводить к искажению их научной сущности. Те элементарные познания об окружающем мире, которые учащиеся получают в младшем возрасте, не должны отвергаться впоследствии, а должны лишь только расширяться и обогащаться. Терминология же должна оставаться единой на протяжении всей учебы.

2. В процессе обучения необходимо знакомить учащихся с новейшими достижениями в соответствующих науках, с происходящими дискуссиями и только что возникшими гипотезами. В доступной форме надо знакомить и с методами научного исследования, т.е. включать учащихся в самостоятельные исследования: проведение наблюдений, постановка экспериментов, работа с литературными источниками, выдвижение соответствующих проблем и их разрешение.

3. Необходимо не только правдивое толкование отдельных явлений, особенно в области общественных наук, но и знакомство учащихся с различными точками зрения на них.

4. В процессе изучения закономерностей развития объективного мира у учащихся должно формироваться научное мировоззрение.

5. В процессе обучения необходимо разоблачение различного рода лженаучных и ошибочных теорий, взглядов и представлений.

На основании вышеперечисленных принципов обучения к системе дидактических игр формирующих первичные представления о параллельных процессах у школьников предъявляются следующие требования:

1. Целевой и содержательный компоненты системы дидактических игр должны соответствовать современному уровню развития научно-технического прогресса.

2. Учебный материал должен быть представлен в доступной для понимания учащимся форме.

3. Применяемые методы обучения должны соответствовать современным требованиям.

4. Дидактические игры должны быть направлены на формирование параллельного стиля мышления.

5. Простота и понятность правил дидактических игр.

6. Каждая игра должна отражать те или иные свойства и особенности параллельных процессов.

#### **Цель и содержание системы дидактических игр.**

На основании требований к системе дидактических игр можно выделить её цель.

**Целью** системы дидактических игр является формирование первичных (интуитивных) понятий, знаний, умений и навыков в области основ параллельных процессов и особенностях совместной работы исполнителей. Эта цель обусловлена следующими положениями. Во-первых,

программирование является интенсивно развивающейся областью, являющейся одним из основных инструментов для решения самых разнообразных задач. Во-вторых, наблюдается стремительный прогресс в области вычислительной техники и программного обеспечения. Многопроцессорные вычислительные системы начинают активно внедряться во все сферы научной и производственной деятельности информационного общества. Меняются вычислительные методы решения задач, совершенствуются технологии программирования. Стремительно развивается использование серверных кластеров и систем распределенных вычислений. Одним словом, параллельные вычисления становятся неотъемлемой частью нашей жизни.[49]

В результате использования системы дидактических игр формирующих первичные представления о параллельных процессах на уроках информатики учащийся должен освоить следующие элементы основ параллельного программирования:

1. Совместная работа нескольких исполнителей и распараллеливание «внутри» одного исполнителя при наличии нескольких обрабатывающих устройств.

2. Виды параллелизма: параллелизм истинный и псевдопараллелизм (один процессор выполняет частями несколько программ).

3. Исполнители однотипные и разнотипные. Работы однотипные и разнотипные.

4. Соотношение «исполнители – работы»: 1 исполнитель – 1 работа, 1 исполнитель – N работ (псевдопараллельное выполнение или истинный параллелизм при наличии нескольких обрабатывающих устройств для разных работ), N исполнителей – 1 работа, N исполнителей – N работ.

5. Выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости выполнения работы от количества

исполнителей. Зависимость стоимости выполнения работы от количества исполнителей.

6. Нелинейное возрастание скорости работы при увеличении количества исполнителей.

7. Оптимальное количество исполнителей для выполнения заданной работы. Оптимальная загрузка исполнителей. Оптимальный порядок действий. Балансировка нагрузки.

8. Механизмы синхронизации исполнителей.

9. Возможность распараллеливания алгоритмов. Возможная степень распараллеливания.

10. Существование алгоритмов, не поддающихся распараллеливанию.

11. Согласование деятельности исполнителей. Виды согласования: по времени и по результатам.

12. Ресурсы. Ресурсы разделяемые и неразделяемые, расходуемые и повторно используемые. Утилизация потребленных ресурсов. Конкуренция исполнителей за ресурсы. Блокировка. Клинч (тупик).

13. Механизмы согласования действий исполнителей.

Представления о вышеперечисленных понятиях, свойствах и процессах формируются последовательным и систематичным введением игровых ситуаций с их последующим обсуждением. Типы игр и порядок их ввода рассмотрен в следующем параграфе.

## **§2. Типы дидактических игр, формирующих представления о параллельных процессах и их место в курсе информатики для 1-6 классов.**

Поскольку одной из особенностей системы дидактических игр является пропедевтика основ параллельного программирования, то все задачи необходимо классифицировать их по содержанию, уровню сложности, длительности и способу проведения.

Нами была проведена такая работа, и были получены следующие классификации игр:

I. Классификация по содержанию и сложности включаемых в игру представлений о параллельных процессах. Вначале в таблице представлены игры с более простыми понятиями и свойствами параллельных процессов, чем ниже вводимое свойство в таблице, тем выше его уровень сложности. (Таблица 1):

*Таблица 1. Классификация дидактических игр по содержанию и сложности включаемых в игру представлений.*

<b>Основные свойства параллельных процессов</b>	<b>Название игры</b>
Разделение работы на части.	Землекопы, стройка, поднимание парты, построение «светофора», построение замков, сетевой график, изготовление браслетов, гримеры.
Однотипные и разнотипные исполнители	Землекопы, экипаж танка, стройка, поднимание парты, построение «светофора», построение замков, сетевой график, изготовление браслетов, гримеры.

Согласование деятельности исполнителей	по времени	Поднимание парты, сетевой график, экипаж танка
	по результатам	Построение «светофора», стройка, сетевой график, экипаж танка
	по ресурсам	Изготовление браслетов
Балансировка исполнителей.	загрузки	Построение замков, проектирование производства, примеры.
Взаимная блокировка		Клинч

II. Классификация по количеству участников и материальному обеспечению. В таблице 2 представлены игры которые разделены на командные и игры которые можно рассматривать как в малой группе, так и индивидуально с каждым учеником.

*Таблица 2. Классификация дидактических игр по количеству участников и материальному обеспечению.*

<b>Число участников</b>	<b>Игры требующие материальное обеспечение</b>	<b>Игры не требующие специального материального обеспечения</b>
1-3	Построение «светофора», изготовление браслетов, танковый экипаж.	Стройка, проектирование производства, примеры, сетевой график.
3-9	Поднимание парты, построение «светофора», изготовление браслетов, построение замков.	Стройка, землекопы, сетевой график.
Более 9	Построение замков, клинч.	

Рассмотрим классифицированные выше игры более подробно и предложим последовательность ввода системы дидактических игр в школьный курс информатики.

Формирование понятий о параллельных процессах стоит начать с самого простой варианта задач – это класс задач на разделение работы на части и совместное выполнение работы однотипными исполнителями.

Рассмотрим задачи, предложенные Босовой Л.Л. в статье «Параллельные алгоритмы для младших школьников» - задача о наборе текста, задача о гримерах и «Директор строительства» [4].

**Задача о наборе текста.** Три школьницы Света, Настя и Марина одинаково быстро и хорошо умеют набирать текст на компьютере. Если любые две из этих девочек будут работать одновременно, то смогут напечатать материалы для школьной газеты за 1 час. Сколько времени они потратят на эту работу, если будут выполнять её вместе?

Задач подобных этой достаточно много в курсе математики. С точки зрения параллельных вычислений здесь важно обратить внимание на места, связанные с возможностью распараллеливания: все хорошо, если девочкам предстоит перепечатать 6 одинаковых по количеству знаков заметок для стенгазеты. А если заметок всего 2? Как делить работу на части? Сколько времени потребуется на «объединение полученных результатов и получение конечного результата»? [4].

**Задача о гримерах.** Три актера готовятся к спектаклю. С ними работают два опытных гримера. Каждый актер должен быть накрашен и причесан. Макияж у каждого актера продолжается 30 минут, а причесывание только 10 минут. Необходимо спланировать работу гримеров таким образом, чтобы актеры как можно быстрее подготовились к выходу на сцену. Сколько для этого потребуется времени?

В этом примере мы также имеем дело с однотипными исполнителями – двумя гримерами, выполняющими одинаковые действия за равное время.

В этой задаче можно ставить дополнительное условие: время ожидания актеров своей очереди должно быть минимальным. Можно манипулировать с числом актеров и гримеров, изменять временные характеристики, вводить других исполнителей (например, костюмеров) и строить на этой основе целый класс задач [4].

Последовательность действий гримеров представлена в таблице 3.

*Таблица 3. Последовательность действий гримеров в задаче о гримерах.*

Время	Гример 1	Гример 2
10 мин		Накрасить (2)
10 мин	Причесать (1)	
10 мин		Причесать (3)
10 мин		
10 мин	Причесать (2)	Накрасить (1)
10 мин		Накрасить (3)

Исполнитель «Директор строительства» (ДС) руководит работой строительных бригад, возводящих здание из блоков. Всякий блок независимо от формы и размера может быть установлен одной бригадой за один день. Две бригады не могут устанавливать один и тот же блок. Установка блока может начаться только после того, как установлены все блоки, на которые он опирается. Строительная бригада устанавливает блок по команде

*ДС установи (n),*

где n – номер блока.

Необходимо построить здание следующей конструкции (рис. 7):

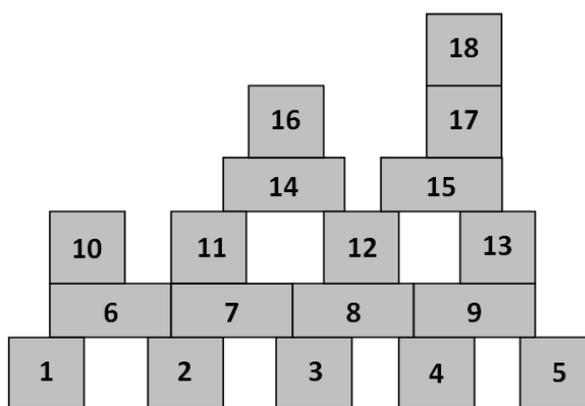


Рис. 7. Чертеж конструкции которую необходимо построить

Необходимо написать алгоритм строительства этого здания за 6 дней тремя бригадами. Для записи алгоритма используется следующая таблица 4.

Таблица 4. Таблица для записи алгоритм строительства в задаче «Директор строительства»

	Бригада 1	Бригада 2	Бригада 3
День 1	установи ( )	установи ( )	установи ( )
День 2	установи ( )	установи ( )	установи ( )
День 3	установи ( )	установи ( )	установи ( )
День 4	установи ( )	установи ( )	установи ( )
День 5	установи ( )	установи ( )	установи ( )
День 6	установи ( )	установи ( )	установи ( )

Подборка задач, подобных этой, представлена в книге [17] – самом значимом достижении отечественной школы в области параллельного программирования для младших школьников. [4].

Можно разнообразить данную задачу и использовать различную вариацию заданий:

- строить здание одной бригадой;
- к строительству привлечь столько бригад, чтобы закончить работу в минимальные сроки;
- за определённое количество дней здание построить с помощью определенного количества бригад;
- доказать, что решение некоторой задачи единственное.

Игровая задача «**Стройка**» подобная исполнителю - «Директор строительства» представлена в приложении 1.

Игра «Стройка» предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
3. Однотипные исполнители (все бригады совершенно равноправны).
4. Однотипные работы.
5. Соотношение «исполнители–работы» –  $N : 1$  ( $N$  бригад выполняют одну общую работу – собирают конструкцию).
6. Согласование деятельности исполнителей. Виды согласования: по частям работы (каждая бригада должна установить свои балки, ни одна балка не должна быть установлена двумя бригадами, каждая балка должна быть кем-то установлена), по времени и по результатам (верхние балки опираются на нижние и могут быть установлены только после нижних).

В игре не отражены следующие моменты:

1. Ресурсы. Ресурсы разделяемые и неразделяемые, расходуемые и повторно используемые. Утилизация потребленных ресурсов.
2. Конкуренция исполнителей за ресурсы. Взаимная блокировка исполнителей.
3. Механизмы согласования действий исполнителей.
4. Псевдопараллельное выполнение процессов (разделение между исполнителями-процессами одного ресурса).
5. Пригодность алгоритмов к распараллеливанию. Возможная степень распараллеливания. Существование алгоритмов, не поддающихся распараллеливанию.

Возможные вариации игры «Стройка» можно посмотреть в приложении 1-5. В приложениях представлены следующие вариации игры:

- написание алгоритма строительства для заданной конструкции с разным количеством строительных бригад;
- изображение конструкции по заданному алгоритму строительства с разным количеством строительных бригад;
- поиск ошибки в заданном алгоритме;
- сравнение стоимости работы строительства нескольких алгоритмов и выбор оптимального алгоритма;

Рекомендуем перед тем как вводить игровую задачу «Стройка», использовать игру «Землекопы», которая предназначена для освоения следующих элементов параллельного программирования:

- Разделение работы на части.
- Однотипные исполнители.
- Однотипные работы.
- Соотношение «исполнители–работы» –  $N : 1$ .
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

В игре игроки изображают землекопов, копающих канаву. Суть игры: продемонстрировать, что большую работу можно выполнить вместе, поделив ее на части между исполнителями.

При разделении работы исполнителей специально допущены ошибки, которые ученикам необходимо будет отследить.

По уровню сложности вводимых элементов основ параллельных вычислений в игру «Землекопы» игра значительно легче, чем игровая задача «Стройка», поскольку в игровой задаче «Стройка» добавляется согласование исполнителей по времени и по результатам работы.

Подробное описание игры «Землекопы» представлено в приложении 6.

Рассмотрим следующую игру «**Поднимание парты**», которую целесообразно вводить при обсуждении разнотипных исполнителей и согласование действия исполнителей по времени.

Игра «Поднимание парты» предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей
- Однотипные работы.
- Исполнители – разнотипные (у каждого своя система команд).
- Согласование по времени.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Играют две бригады по два человека.

Суть игры: две бригады поднимают стол (парту) и ставят на другой стол (парту), при этом каждая бригада исполнителей имеет свою систему команд.

Обсуждение в игре: почему в первые разы поднять стол не удалось, а в последний – удалось? В чем причина? Ответ: причина в том, что в последнем случае обе бригады стали поднимать одновременно. То есть действия бригад были согласованы по времени.

Подробное описание игры «Поднимание парты» представлено в приложении 7.

Рассмотрим следующую игру - **построение «светофора»**.

Задача предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей.
- Однотипные работы.
- Однотипные исполнители.
- Согласование по результатам.
- Соотношение «исполнители–работы» – N : 1.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

В игре принимают участие 3 человека. Суть игры: надо сложить из кубиков «светофор» из трех рядов одинаковой длины: снизу – зеленый, в середине – желтый, сверху – красный.

Обсуждение: Ряды кубиков будут, скорее всего, сложены всеми игроками примерно за одно и то же время. Но поле этого «желтому» игроку придется ждать, пока свой ряд уложит «зеленый», а «красному» – пока будут уложены оба нижних ряда. (Большая длина ряда делает время ожидания более заметным.)

Здесь – отличие как от деления работы на части в игре «Землекопы» (приложение 6), так и от согласования по времени в игровой задаче «Стройка» (приложение 1-5). В первой игре каждый из землекопов «копал» свою часть канавы независимо от других. Его совершенно не интересовало, какие другие участки уже выкопаны, какие – нет.

Во второй игре требовалась синхронность действий двух бригад.

Подробное описание игры - построение «светофора» представлено в приложении 8, так же в приложении 8 представлен вариант данной игры без использования кубиков.

Рассмотрим следующую игру **«Изготовление браслетов»**, которую целесообразно использовать при вводе понятия ресурс и согласования действия исполнителей по ресурсам.

Подробное описание игры «Изготовление браслетов» представлено в приложении 9.

Задача предназначена для освоения следующих элементов параллельного программирования:

- Согласование по ресурсам.
- Соотношение «исполнители–работы» – 1 : 1.
- Псевдопараллелизм (разделение между исполнителями одного ресурса).
- Разнотипные работы.

- Однотипные исполнители.

Играют 3 человека. Задание: каждый из игроков должен изготовить себе бумажный браслет: вырезать его листа бумаги, раскрасить фломастером (рисунок может быть любым) и склеить.

Чтобы при вырезании одного браслета ножницы были заняты большее время, пусть края браслетов будут фигурными.

Суть игры в том, что выделенных ресурсов недостаточно для одновременной независимой работы всех троих игроков. Поэтому кому-то постоянно придется ожидать, пока другие не закончат некоторые действия и не освободят ресурс. Именно поэтому даются ОДНИ ножницы, ОДИН лист бумаги, ОДИН клей. Нехватку этих ресурсов можно противопоставить фломастерам. Их – три. Поэтому на стадии раскрашивания – в отличие от остальных стадий – все три игрока могут действовать одновременно.

Можно попробовать сыграть с двумя фломастерами. Тогда и этого ресурса будет не доставать.

При обсуждении игры вводится понятие «ресурс» (если дети его еще не знают, в некоторых школах оно может быть введено ранее в курсе экономики или в экономической части курса «Окружающий мир»). Обсуждается вопрос о нехватке ресурсов и вынужденных потерях времени на их ожидание.

В процессе обсуждения игры может возникнуть мысль о том, что изменение порядка действий может привести к сокращению времени выполнения упражнения. Это – так. Но это не есть предмет данной игры.

Главным образом потому, что в этом упражнении нет четкой системы команд исполнителя и счетчика времени. Поэтому измерить ускорение, полученное в результате изменения порядка действий, невозможно. Этот аспект будет подробно обсуждаться в следующей игре «Постройка замков» (приложение 10).

## «Построение замков»

Задача предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей.
- Однотипные работы.
- Однотипные исполнители.
- Соотношение «исполнители–работы» –  $N : 1$ .
- Загрузка исполнителей.
- Ускорение работы при увеличении количества исполнителей.
- Непропорциональность ускорения.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Игроки:

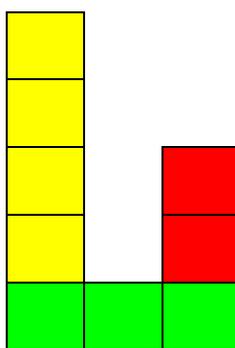
I. Три бригады строителей: две по три человека и одна из двух человек;

II. Три наблюдателя, которые должны отследить правильность действий строительных бригад и срок окончания строительства каждой из бригад;

III. «Хранитель времени», который «отбивает» такты работы и считает их. В этой игре необходима синхронизация: за один такт работы каждый исполнитель выполняет ровно одну команду.

В игре присутствует система команд исполнителей.

Задача каждой из бригад из девяти кубиков построить замок следующего вида (рис. 8):



*Рис. 8. Пример постройкой из кубиков для игры «Построение замков»*

В игре необходимо обсудить разницу в скорости работы двух трехчеловечных бригад.

Обратить внимание на то, что при работе всех бригад обязательно есть хотя бы один простой (пропуск хода). Это простой вызван разной высотой башен. При достраивании более высокой башни на последнем ходе в любом случае работает только один строитель.

Подробное описание игры «Построение замков» представлено в приложении 10.

Значительно большим потенциалом с точки зрения параллельного программирования обладают задачи «Проектирование производства» [18] и «Танковый экипаж» [23], предложенные коллективом пермских авторов.

#### **Задача «Проектирование производства».**

1. Совместная работа нескольких исполнителей.
2. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
3. Однотипные исполнители.
4. Однотипные работы.
5. Соотношение «исполнители–работы» –  $N : 1$ .
6. Оптимальный порядок действий. Оптимальная загрузка исполнителей.

Предприятие должно в минимальный срок изготовить партию приборов (40 штук). Изготовление состоит из 2-х операций: монтаж и калибровка. Монтаж может вестись на 3-х участках. На двух из них установлено новое оборудование, производящее один прибор за 35 час. На третьем стоит старое оборудование. На нем производство одного прибора требует 42 час., и после каждых 3 изделий требуется профилактика оборудования продолжительностью 42 час. Калибровка проводится на одном из двух стендов и требует 28 час. на прибор. Все смонтированные приборы сначала складываются в специальный накопитель, а уже из него поступают на тот или другой стенд. Все монтажные участки начинают работать одновременно. Производство ведется круглосуточно и непрерывно. Требуется спланировать: время начала и конца работы каждого из участков и калибровочных стендов, которые обеспечат скорейшее выполнение заказа, и требуемый объем накопителя (максимальное количество приборов, которое придется в нем разместить). Простои должны быть минимизированы [20].

**Игровая обучающая компьютерная программа «Танковый экипаж»** предложенная коллективом пермских авторов. [23] Подробное описание игры «Танковый экипаж» представлено в приложении 12.

Игра предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
3. Исполнители – разнотипные (у каждого своя система команд).
4. Соотношение «исполнители–работы» –  $N : 1$  (весь экипаж в целом решает одну боевую задачу).
5. Согласование деятельности исполнителей. Виды согласования: по времени и по результатам (для выстрела танк должен занять нужную позицию, орудие должно быть заряжено; в задание специально включена

ситуации, когда командиру и водителю приходится ждать перезарядки орудия).

6. Оптимальный порядок действий.

В игре не отражены следующие моменты:

1. Выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости работы от количества исполнителей. Зависимость стоимости работы от количества исполнителей. Нелинейный рост скорости работы при росте количества исполнителей. Критический путь. Оптимальное количество исполнителей. Оптимальная загрузка исполнителей.

2. Ресурсы. Ресурсы разделяемые и неразделяемые, расходуемые и повторно используемые. Утилизация потребленных ресурсов.

3. Конкуренция исполнителей за ресурсы. Блокировка. Клинч (тупик).

4. Механизмы согласования действий исполнителей.

5. Псевдопараллельное выполнение процессов на компьютере.

6. Пригодность алгоритмов к распараллеливанию. Возможная степень распараллеливания. Существование алгоритмов, не поддающихся распараллеливанию.

Обратим внимание на терминологические сложности, возникающие при обсуждении игры. Во-первых, не хватает термина, для обозначения всей системы «Танковый экипаж». В случае последовательных алгоритмов мы говорим об «исполнителе». Но в случае совместной деятельности нескольких членов экипажа (или нескольких бригад в игре «Стройка») термин «исполнитель» уже некорректен. Здесь действует группа исполнителей.

Во-вторых, непонятно, что считать программой: описание совместной деятельности всех трех членов экипажа, или каждого из них по отдельности. То есть решением задачи является одна – параллельная – программа или три отдельных программы? вопрос остается открытым.

Далее после рассмотрения обучающей компьютерной программы «Танковый экипаж» стоит рассмотреть игровую задачу «Сетевой график», которая включает в себя большое количество элементов параллельных процессов, также прогнозирование результатов выполнения алгоритма.

Игровая задача предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости выполнения работы от количества исполнителей.
3. Однотипные исполнители (все бригады совершенно равноправны).
4. Однотипные работы.
5. Соотношение «исполнители–работы» –  $N : 1$ .
6. Оптимальный порядок действий. Оптимальная загрузка исполнителей.
7. Согласование деятельности исполнителей по времени и по результатам.
8. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

В задаче дан сетевой график, к которому требуется:

1. Изобразить (схематически) сооружение, которое будет построено.
2. Определить, сколько дней потребуется для строительства, если работать будет ровно одна бригада.
3. Определить, сколько дней потребуется для строительства, если работать будет две бригады.
4. Определить, сколько дней потребуется для строительства, если работать будет четыре бригады.
5. Изобразить (схематически) как будет выглядеть стройка через 10 дней после начала строительства, если работать будут 4 бригады.

Подробное описание игры «Построение замков» представлено в приложении 13.

Рассмотрим следующую игру **«Параллельные процессы: Клинч»**.

Данную игру целесообразно использовать при рассмотрении конкуренции процессов за ресурсы, при вводе понятия – такт работы программы, так же при рассмотрении взаимной блокировки параллельных процессов.

Задача предназначена для освоения следующих элементов параллельного программирования:

- Взаимная блокировка параллельных процессов.
- Соотношение «исполнители–работы» –  $N : N$ .
- Разнотипные работы.
- Однотипные исполнители.
- Псевдопараллелизм.
- Конкуренцию процессов за ресурсы.

Участники делятся на 3-4 бригады.

Каждая бригада должна составить алгоритм для следующего процесса:  
Подойти к столу. Из бутылки перелить в кастрюлю 2 кружки воды из бутылки №1 и 2 кружки из бутылки №2. Вернуться на место.

В задаче имеется система команд исполнителей.

Бригада записывает алгоритм на листе флип-чарта.

Все листы с алгоритмами вывешиваются рядом.

Каждая бригада выставляет Исполнителя и Командера (указателя команд). Командер будет зачитывать команды из алгоритма, Исполнитель выполнять их. Также существует Диспетчер, который руководит всеми Командерами.

Смысл игры: продемонстрировать конкуренцию процессов за ресурсы.

Взаимная блокировка процессов не являются обязательным состоянием в игре, однако она имеет место быть.

Ради этого в систему команд включены некоторые уточнения. Например, требование наливать воду именно в пустую кружку, нумерация бутылок (если рядом со столом стоит еще какая-то бутылка, она не может быть задействована, поскольку не имеет номера).

Хорошо дождаться ситуации, когда все бригады заблокируют друг друга и прокрутить несколько циклов, когда ни одна бригада не сможет выполнить свою команду.

Формально слабым местом игры м.б. тот факт, что моделируется выполнение нескольких процессов на одном процессоре, в то время, как современные ЭВМ – многоядерные. Но это слабость чисто формальная. В реальности количество процессов может многократно превышать количество ядер. Поэтому ситуация, когда несколько процессов выполняются на одном процессоре – вещь совершенно обыденная. Можно позволить нескольким бригадам действовать одновременно. Это только запутает картину, сделает ее более смазанной.

Можно экспериментировать с количеством ресурсов. Взять 2 кружки и одну бутылку, 2 бутылки и 1 кружку, 2 бутылки и 2 кружки. Заметим, что наличие двух бутылок дает бригадам некоторую свободу. Вдруг кто-то начнет не с бутылки №1, а с бутылки №2.

Можно экспериментировать с дисциплиной обслуживания:

1. Все процессы имеют одинаковый приоритет. Период (количество тактов) постоянный. Величина периода определяется кубиком.

2. Для каждого процесса разыгрывается его приоритет и время активации. Процесс со старшим приоритетом встает в очередь первым.

3. Процесс со старшим приоритетом прерывает работу процесса с младшим приоритетом.

Подробное описание игры «Параллельные процессы: Клинч» представлено в приложении 11.

В данном параграфе были рассмотрены дидактические игры и их классификации, которые могут способствовать формированию первичных представлений учащихся о параллельных процессах.

Апробация системы дидактических игр не проводилась.

## **Выводы по главе 2**

Во второй главе определены ведущие дидактические принципы обучения школьников основам параллельных процессов при помощи системы дидактических игр, такие как принцип научности, доступности, наглядности, системности и систематичности. Сформулированы цели обучения школьников основам параллельных процессов, где основной целью является формирование первичных (интуитивных) понятий, знаний, умений и навыков в области основ параллельных процессов и особенностях совместной работы исполнителей.

В работе описано основное содержание системы дидактических игр, представлены их классификации по двум признакам: по содержанию и сложности включаемых в игру представлений и количеству участников и материальному обеспечению.

Приведены примеры разработанных игр и заданий, также предложены методические рекомендации к организации игр, порядку их ввода, приведены скрытые в играх понятия и особенности параллелизма.

## **Заключение**

В заключение приведём результаты и выводы, полученные в ходе работы.

1. В результате проделанной работы был проведен анализ литературы с целью выявления понятийных проблем линии алгоритмизации в школьном курсе информатики.

2. Определено понятие «параллельного стиля мышления» и выделены этапы его формирования, также описана роль дидактических игр в формировании параллельного стиля мышления.

3. Определено базовое содержание и принципы обучения основам суперкомпьютерных технологий и предложена последовательность изучения данной темы в 1-6 классах.

4. В рамках деятельностного подхода построена система дидактических игр, которая может способствовать формированию представлений о параллельных процессах и создающая предпосылки к формированию параллельного стиля мышления.

5. Для системы игр приведены некоторые методические рекомендации по организации, порядку введения, особенностям содержания игр.

## Библиографический список

1. Алексашкина Л.Н. Деятельностный подход в изучении истории в школе// Преподавание истории и обществознания в школе. 2005. – № 9.
2. Босова Л.Л. «Информатика: учебник для 7 класса», изд. «БИНОМ», 2008.
3. Босова Л.Л. «Как правильно выстроить непрерывный курс школьной информатики» [http://infojournal.ru/e\\_bosova.htm#13](http://infojournal.ru/e_bosova.htm#13)
4. Босова Л.Л. Параллельные алгоритмы в начальной и основной школе. //Информатика в школе. 2015, №2. С.24-27.
5. Босова Л.Л., Босова А.Ю. «Уроки информатики в 5-6 классах. Методическое пособие», изд. «БИНОМ», 2006.
6. Воеводин В.В. Вычислительная математика и структура алгоритмов: 10 лекция о том, почему трудно решать задачи на вычислительных системах параллельной архитектуры и что надо знать дополнительно, чтобы успешно преодолевать эти трудности: учебник. М.: Изд-во МГУ 2010.
7. Вольхин К.А., Пак Н.И. О состоянии графической подготовки учащихся в школе с позиции информационного подхода // Вестник КГПУ им. В.П. Астафьева. 2011. № 3 (17). Т. 1: Психолого-педагогические науки. С. 74-79 с.
8. Гаврилова И.В. Первое путешествие в «параллельный мир». //Информатика в школе. 2015, №6. С.16-19.
9. Газейкина А.И., Стили мышления и обучение программированию студентов педагогического вуза /– 2006.
10. Гальперин П.Я. Введение в психологию: Учеб. пособие для вузов. 4-е изд. - М.:Университет, 2000. - 336 с.
11. Голуб Б.А. Основы общей дидактики : учеб. пособие для студ. педвузов / Б.А. Голуб. – М. : ВЛАДОС, 1999. – 96 с.

12. Городняя Л.В. статья «О курсе «Начала параллелизма» для школьников» Институт систем информатики им. А.П. Ершова СО РАН Новосибирск, Россия.

13. Дитер М.Л., Плаксин М.А. Параллельные вычисления в школьной информатике. Игра «Стройка». //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т. – Пермь, 2014. – С.258-261.

14. Дружинин В. Н. Экспериментальная психология: Учебник для вузов /. – 2-е изд., доп. – СПб.: Питер, 2003. – 319 с: ил. – (Серия “Учебник для вузов”)

15. Занков Л.В. Избранные педагогические труды / Л.В. Занков. – М.: Педагогика, 1990. – 424 с.

16. Занков Л.В., Система общего развития школьников, URL: <https://www.o-detstve.ru/forteachers/primaryschool/educprocess/120.html> (дата обращения: 06.05.2016).

17. Звонкин А.К., Кулаков А.Г., Ландо С.К., Семенов А.Л., Шень А.Х., Алгоритмика: 5-7 классы: Учебник и задачник для общеобразоват. учебных заведений — М.: Дрофа, 1996.

18. Иванова Н.Г., Плаксин М.А., Русакова О.Л. Конкурс «ТРИЗформашка» как площадка для апробации заданий на параллельное программирование. // Информатика в школе: прошлое, настоящее и будущее: материалы Всеросс. научн.-метод. Конф. По вопросам применения ИКТ в образовании, 6–7 февраля 2014 г.; Перм. гос. нац. иссл. ун-т. – Пермь, 2014. С 233–236.

19. Игра в жизни ребенка. URL: <http://www.psi.lib.ru/detsad/raznoe/igvgr.htm> (дата обращения: 29.05.2016).

20. Киселева Е.Ю. Потенциал суперкомпьютерной тематики в проектно-исследовательской деятельности учащихся. //Информатика в школе. 2015, №2. С.20-23
21. Киселева Е.Ю. Реконструкция параллельной модели вычислений на примере задачи суммирования чисел. //Информатика в школе. 2013, №10.
22. Коменский Я.А. Великая дидактика. Избр. пед. соч. : Т. 1. / Я.А. Коменский – М. : Педагогика, 1982. –476 с.
23. Кучев А.Д., Плаксин М.А. Параллельные вычисления в школьной информатике. Игра «Танковый экипаж». //Информатика в школе: прошлое, настоящее и будущее.: материалы Всеросс. науч.-метод. конф. по вопросам применения ИКТ в образовании, 6-7 февраля 2014 г. /Перм. гос. нац. иссл. ун-т. — Пермь, 2014. — С.241-243.
24. Ландо С.К. «Информатика 6. Книга для учителя», изд. «Просвещение», 2006 г.
25. Лапчик М.П. / Теория и методика обучения информатике: учебник / М.: Академа, 2008. 585 с.
26. Леонтьев А.Н. Деятельность. Сознание. Личность. М.: Смысл; Академия, 2004. 352 с.
27. Логинов А.В. Исторические предпосылки и перспективы суперкомпьютерного образования в школьном курсе информатики. //Информатика в школе. 2015, №2. С.17-19.
28. Макарова Н.В. «Информатика и ИКТ практикум. 7-9 класс. Базовый уровень», изд. «Питер», 2007 г.
29. Материалы Летней Суперкомпьютерной Академии. <http://academy.hpc-russia.ru/>. [Электронный ресурс] Проверено 20.05.2016.
30. Мэттсон Т., Чернышев А. Введение в технологии параллельного программирования. URL: <http://software.intel.com/ru-ru> (дата обращения: 05.03.2016)

31. Общая педагогика URL: [http://krip.kbsu.ru/pd/did\\_lect\\_4.html#p6](http://krip.kbsu.ru/pd/did_lect_4.html#p6)  
(дата обращения: 29.04.2016)
32. Пак Н.И. Обучение разума как информационный процесс: сб. тезисов Российско-корейской научной конференции, Звенигород, 2011. С. 81—83.
33. Пирогов Н. Вопросы жизни. Соч. / Н. Пирогов. : – СПб, 1887. – 116 с.
34. Пискунов А. И. Хрестоматия по истории зарубежной педагогики. - М. : Просвещение, 1981. - С. 115 - 119.
35. Подласый И.П. Педагогика: 100 вопросов - 100 ответов: учеб. пособие для вузов - М.: ВЛАДОС-пресс, 2004. - 365 с.
36. Посмакова Н.М. Значение игры для ребенка // Психология, социология и педагогика. 2013. № 1 [Электронный ресурс]. URL: <http://psychology.snauka.ru/2013/01/1703> (дата обращения: 28.05.2016).
37. Рубинштейн С. Л. Основы общей психологии. СПб.: Питер, 2008. 713 с.
38. Семакин И.Г., Залогова Л.А. и др. «Информатика. Базовый курс. 7-9 классы», изд. «БИНОМ», 2003 г.
39. Сокольская М.А. О методах обучения основам параллельного программирования будущих учителей информатики. // Инновации в непрерывном образовании, 2012. №4.
40. Сокольская М.А. О содержании специального курса «Введение в параллельное программирование»././ Инновационные процессы в современном образовании России как важнейшая предпосылка социально-экономического развития общества: сборник докладов региональной научно-практической конференции. Красноярск: СФУ, 2009. С. 123 – 125.
41. Сокольская М.А., Степанова Т.А., Уточнение понятия «Параллельный стиль мышления» на основе информационно-

деятельностного подхода / Вестник Красноярского государственного педагогического университета им. В.П. Астафьева, № 1 / 2012.

42. Суперкомпьютерные конфигурации «СКИФ» / Мн.:ОИПИ НАН Беларуси, 2005.-195 с. URL: <http://skif.pereslavl.ru/psi-info/rcms-skif/skif-publications/2005-rus/book-vers2.pdf> (дата обращения: 21.05.2016).

43. Сухомлинский В.А. Сердце отдаю детям. — Киев: Радянська школа, 1974 г. - 288 с.

44. Турушев М.И. Методика обучения приемам робототехники и легоконструирования с применением идей параллелизма: сборник докладов параллелизма Молодежь и наука XXI века: материалы XIII Всероссийской (с международным участием) научно-практической конференции студентов, аспирантов и молодых ученых. В 4 томах. Том 1. Красноярск, 17 апреля 2012 г. / отв. за выпуск В.И. Пихутина; ред. кол. Краснояр. гос. пед. ун -т им. В.П. Астафьева. – Красноярск, 2012. – 307 с.

45. Турушев М.И. О возможностях использования принципов взаимодействия процессов в легоконструировании: сборник докладов 50-й Международной научной студенческой конференции «Студент и научно-технический прогресс»: Информационные технологии / Новосиб. гос. ун-т. Новосибирск, 2012. 230с.

46. Угринович Н.Д. «Информатика. Базовый курс 9», изд. «БИНОМ», 2006.

47. Федеральный государственный образовательный стандарт основного общего образования / М-во образования и науки Рос. Федерации. – М. : Просвещение, 2011. – 48 с., с.9

48. Что такое CUDA URL: [http://www.nvidia.ru/object/what\\_is\\_cuda\\_new\\_ru.html](http://www.nvidia.ru/object/what_is_cuda_new_ru.html) (дата обращения: 05.05.2016).

49. Шишкин В.А., Методика визуализированного обучения параллельному программированию студентов педагогического университета / Красноярск, 2014.

50. Энциклопедии и словари URL: [http://enc-dic.com/enc\\_big/Propedevtika-48425.html](http://enc-dic.com/enc_big/Propedevtika-48425.html) (дата обращения: 05.05.2016).

51. Юдин Э. Г. Методология науки. Системность. Деятельность. М.. 1997.

## ПРИЛОЖЕНИЕ 1

### Игровая задача «Стройка»

Задача предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
3. Однотипные исполнители (все бригады совершенно равноправны).
4. Однотипные работы.
5. Соотношение «исполнители–работы» –  $N : 1$  ( $N$  бригад выполняют одну общую работу – собирают конструкцию).
6. Согласование деятельности исполнителей. Виды согласования: по частям работы (каждая бригада должна установить свои балки, ни одна балка не должна быть установлена двумя бригадами, каждая балка должна быть кем-то установлена), по времени и по результатам (верхние балки опираются на нижние и могут быть установлены только после нижних).

Игровая задача «Стройка» устроена следующим образом.

Дан чертеж конструкции, которую надо собрать из горизонтальных и вертикальных балок. Для простоты балки перенумерованы. Вертикальные балки ставятся на край или середину нижележащей горизонтальной балки. Горизонтальные балки можно уложить на землю (между двумя указанными точками), на одну вертикальную балку (серединой) или на две вертикальные балки (концами).

За один рабочий день одна бригада способна установить ровно одну балку (в любую позицию, в любое положение, но ровно одну).

Стройку ведут две, три или четыре бригады, работающие одновременно. Бригады абсолютно равнозначны, могут выполнять одни и те же действия за одно и тоже время.

Задание игроку заключается в том, чтобы написать программу для совместной работы заданного числа бригад такую, чтобы их совместная работа привела к сборке заданной конструкции.

Каждая бригада действует по своей программе, не глядя на другие бригады. Все программы стартуют одновременно.

Точки на стройплощадке обозначаются буквами (А, В,..).

Система команд исполнителя (бригады строителей):

*Таблица5. Система команд исполнителя для игровой задачи «Стройка»*

<b>Полная форма записи</b>	<b>Краткая форма записи</b>
Уложить (N) (точка-1, точка-2)	Улож (N) (А,В)
Уложить (N) краями на (балка-1, балка-2)	Улож (N) (М,К)
Уложить (N) серединой на (М)	Улож (N) (М)
Установить (N) на левый край (М)	Уст (N) лев (М)
Установить (N) на правый край (М)	Уст (N) прав (М)
Установить (N) на середину (М)	Уст (N) сред(М)
Пауза (Пропуск хода)	Пауза

N, М, К – номера балок.

А, В – точки на стройплощадке.

Пример конструкции, для построения которой должен быть составлен алгоритм, приведен на рис. 9.

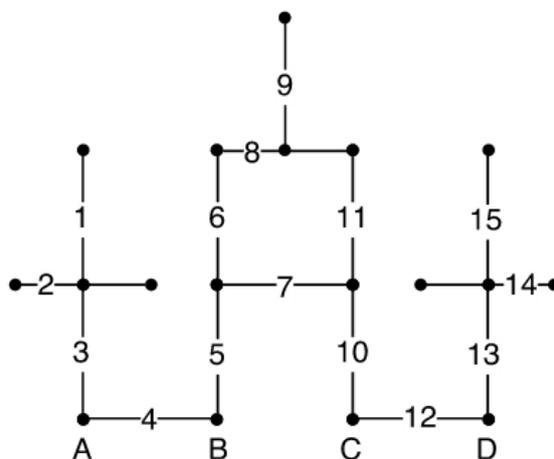


Рис. 9. Чертеж конструкции, для построения которой должен быть составлен алгоритм в игровой задаче «Стройка»

Решение (одно из возможных):

Количество бригад: 3.

Таблица 6. одно из возможных решений игровой задачи «Стройка» для трех бригад

№ хода	Бригада 1	Бригада 2	Бригада 3
1.	Улож (4) (A,B)	Улож (12) (C,D)	Пауза
2.	Уст (3) лев (4)	Уст (5) прав (4)	Уст (10) лев (12)
3.	Улож (2) (3)	Улож (7) (5,10)	Уст (13) лев (12)
4.	Уст (6) лев (7)	Уст (11) прав (7)	Улож (14) сред (13)
5.	Уст (1) сред (2)	Улож (8) (6,11)	Уст (15) сред (14)
6.	Пауза	Уст (9) сред (8)	Пауза

В безмашинном режиме игра может использоваться для обсуждения следующих вопросов: Выполнение одной и той же работы одним

исполнителем и группой исполнителей. Зависимость скорости работы от количества исполнителей. Зависимость стоимости работы от количества исполнителей. Нелинейный рост скорости работы при росте количества исполнителей. Критический путь. Оптимальное количество исполнителей. Оптимальная загрузка исполнителей. Оптимальный порядок действий.

Игра: «Стройка»

Задание: Выполнить алгоритм.

Бригада 1:

1. Установить балку 1 в точку А.
2. Уложить балку 5 левым концом на балку 1, правым на балку 2.
3. Установить балку 8 на правый край балки 5.
4. Уложить балку 10 левым концом на балку 8, правым на балку 9.

Бригада 2:

1. Установить балку 2 в точку В.
2. Уложить балку 6 левым концом на балку 2, правым на балку 3.
3. Установить балку 9 на правый край балки 6.

Бригада 3:

1. Установить балку 3 в точку С.
2. Установить балку 4 в точку D.
3. Уложить балку 7 левым концом на балку 3, правым на балку 4.

Что получилось (рис. 10)?

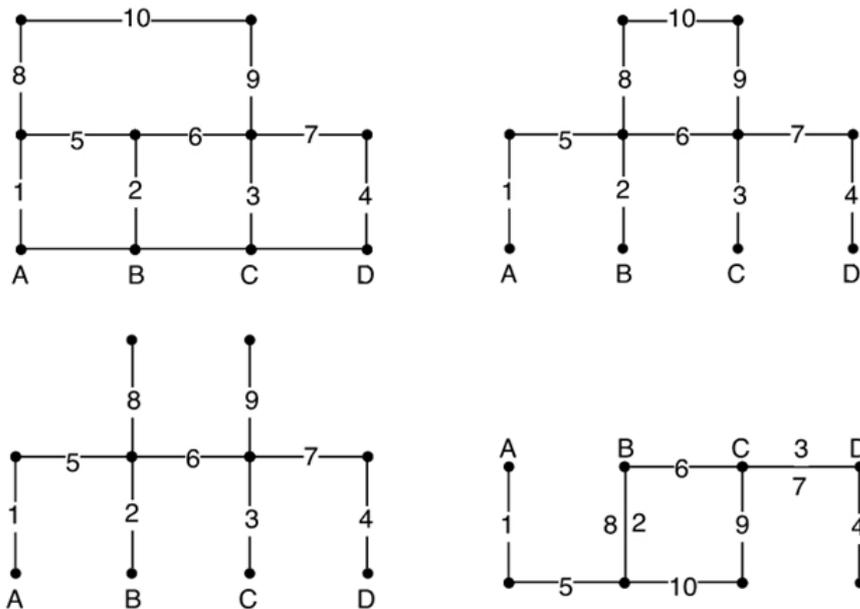


Рис. 10. Чертежи конструкции, из которых нужно выбрать одну подходящую по условию данного алгоритма

### ПРИЛОЖЕНИЕ 3

#### Игра: «Стройка»

Задание: Составить алгоритм программы для совместной работы заданного числа бригад, такую, чтобы их совместная работа привела к сборке заданной конструкции (рис 11).

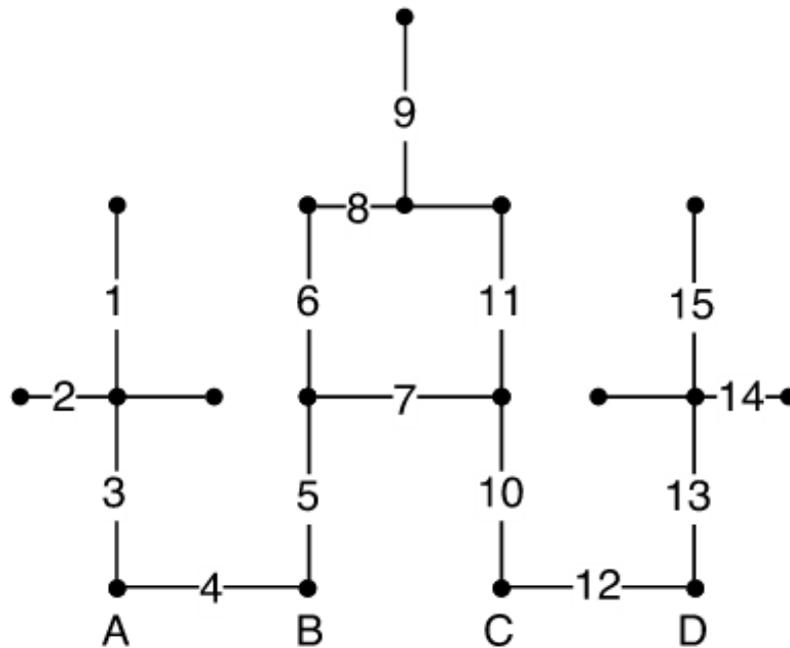


Рис. 11. Чертеж конструкции, для построения которой должен быть составлен алгоритм

Игра: «Стройка»

Задание: Найти ошибку в алгоритме.

Проанализировать алгоритмы программ, приведенные для строительства данной конструкции (рис. 12) и найти ошибки в алгоритме.

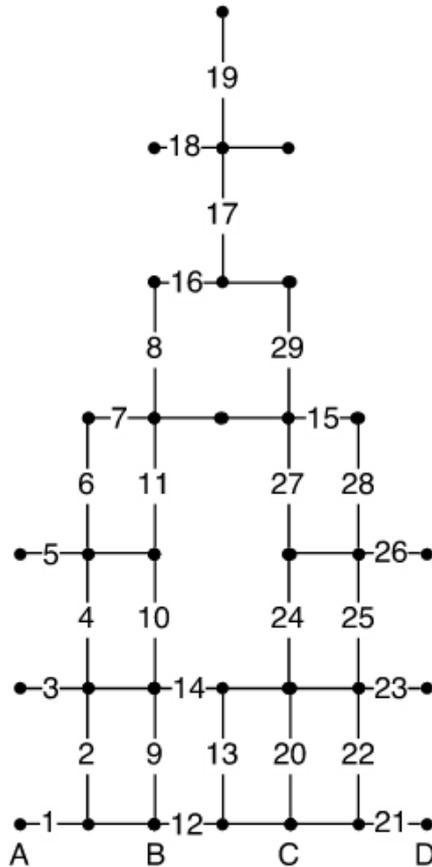


Рис. 12. Чертеж конструкции которую необходимо построить

**Алгоритм № 1**

**Бригада 1**

1. Уложить (1) между (А,В)
2. Установить (2) на середину (1)
3. Уложить (3) серединой на (2)
4. Установить (4) на середину (3)
5. Установить (10) справа на (3)
6. Уложить (5) серединой на (4)

**Алгоритм № 2**

**Бригада 1**

1. Уложить (1) между (А,В)
2. Установить (2) на середину (1)
3. Уложить (3) серединой на (2)
4. Установить (4) на середину (3)
5. Уложить (5) серединой на (4)
6. Установить (6) на середину (5)

7. Установить (6) на середину (5)
8. Уложить (7) серединой на (11)
9. Установить (8) на середину (7)
10. Пауза
11. Пауза
12. Пауза
13. Пауза
14. Пауза

### **Бригада 2**

1. Уложить (12) между (В,С)
2. Установить (9) справа на (1)
3. Установить (20) справа на (12)
4. Уложить(14) серединой на (13)
5. Установить (24) справа на (14)
6. Пауза
7. Установить (11) справа на (5)
8. Пауза
9. Пауза
10. Установить (29) на среди ну (15)
11. Уложить (16) концами на (8,29)
12. Установить (17) на среда ну (16)
13. Уложить(18) серединой на (17)
14. Установить (19) на середину (18)

### **Бригада 3**

1. Уложить (21) между (С,Д)
2. Установить (13) на середину (12)
3. Установить (22) на середину (21)
4. Уложить (23) серединой на (22)
5. Установить (25) на середину (23)

7. Уложить (7) серединой на (11)
8. Установить (8) на середину (7)
9. Установить (29) на середину (15)
10. Пауза
11. Пауза
12. Пауза

### **Бригада 2**

1. Уложить (12) между (В,С)
2. Установить (9) справа на (1)
3. Уложить (14) серединой на (13)
4. Установить (10) справа на (3)
5. Установить (24) справа на (14)
6. Установить (11) справа на (5)
7. Уложить (26) серединой на (25)
8. Установить (28) на середину (26)
9. Уложить (16) концами на (8,29)
10. Установить (17) на середину (16)
11. Уложить (18) серединой на (17)
12. Установить (19) на середину (18)

### **Бригада 3**

1. Уложить (21) между (С,Д)
2. Установить (13) на середину (12)
3. Установить (20) справа на (12)
4. Установить(22) на середину (21)
5. Уложить (23) серединой на (22)
6. Установить (25) на середину (23)
7. Установить (27) слева на (26)
8. Уложить (15) серединой на (27)
9. Пауза

- |                                     |          |
|-------------------------------------|----------|
| 6. Уложить (26) серединой на (25)   | 10.Пауза |
| 7. Установить (27) слева на (26)    | 11.Пауза |
| 8. Установить (28) на середину (26) | 12.Пауза |
| 9. Уложить (15) серединой на (27)   |          |
| 10.Пауза                            |          |
| 11.Пауза                            |          |
| 12.Пауза                            |          |
| 13.Пауза                            |          |
| 14.Пауза                            |          |

### **Алгоритм № 3**

#### **Бригада 1**

1. Уложить (1) между (А,В)
2. Установить (2) на середину (1)
3. Установить (9) справа на (1)
4. Установить (20) справа на (12)
5. Уложить (3) серединой на (2)
6. Установить (4) на середину (3)
7. Установить (10) справа на (3)
8. Уложить (5) серединой на (4)
9. Установить (6) на середину (5)
10. Установить (11) справа на (5)
11. Уложить (7) серединой на (11)
12. Установить (8) на середину (7)
13. Пауза
14. Пауза
15. Пауза
16. Пауза
17. Пауза

## **Бригада 2**

1. Уложить (12) между (В,С)
2. Уложить (21) между (С,Д)
3. Установить (13) на середину (12)
4. Установить (22) на середину (21)
5. Уложить (14) серединой на (13)
6. Уложить (23) серединой на (22)
7. Установить (24) справа на (14)
8. Установить (25) на середину (23)
9. Уложить (26) серединой на (25)
10. Установить (27) слева на (26)
11. Установить (28) на середину (26)
12. Уложить (15) серединой на (27)
13. Установить (29) на середину (15)
14. Уложить (16) концами на (8,29)
15. Установить (17) на середину (16)
16. Уложить (18) серединой на (17)
17. Установить (19) на середину (18)

Игра: «Стройка»

Задание: Проанализировать алгоритмы.

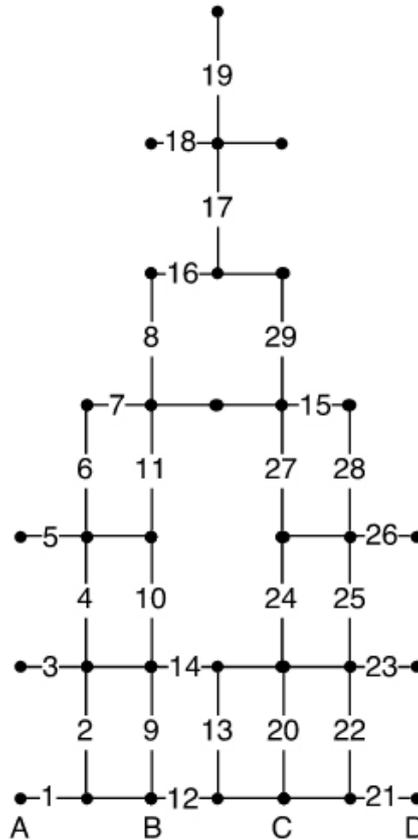


Рис. 13. Чертеж конструкции которую необходимо построить

1. Сравните сроки строительства по двум оставшимся алгоритмам из задачи (приложение 4)?
2. Сравните стоимость строительства по этим алгоритмам?
3. В какой момент и сколько бригад следовало бы дополнительно нанять на работу или наоборот уволить для того, чтобы не увеличивать срок строительства по каждому из алгоритмов, но уменьшить его стоимость? Сколько денег удалось бы сэкономить?
4. Пусть использование построенного сооружения приносит прибыль в одну монету в день. Какой из алгоритмов строительства выгодней, если все бригады нанимаются на весь срок строительства? Почему? Какой из алгоритмов выгодней, если можно нанимать и увольнять бригады по мере надобности? Почему?

### Землекопы

Задача предназначена для освоения следующих элементов параллельного программирования:

- Разделение работы на части.
- Однотипные исполнители.
- Однотипные работы.
- Соотношение «исполнители–работы» –  $N : 1$ .
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Играют 3 человека.

Оборудование: доска, три куска мела (лучше – разноцветных), карточки с заданиями для каждого игрока.

Игроки изображают землекопов, копающих канаву. Канаву будет изображать линия на доске. Лучше, если у каждого землекопа линия будет своего цвета.

Учитель на доске мелом ставит 10 точек, отмеченных буквами (от А до И), и раздает землекопам карточки с заданиями. Задания выглядят так:

- a) Провести: АБ, ГД, ЁЖ.
- b) Провести: ГД, ЕЁ, ЗИ.
- c) Провести: БВ, ДЕ, ЖЗ.

Суть игры: продемонстрировать, что большую работу можно выполнить вместе, поделив ее на части между исполнителями.

При разделении работы специально допущены ошибки: участок ВГ не распределен никому, зато участок ГД распределен дважды (т.е. канава там получится то ли вдвое глубже, то ли вдвое шире). Это – те ошибки, которые необходимо отслеживать при распределении работы: каждая часть работы должна быть поручена одному из исполнителей, каждая часть должна быть поручена ровно одному исполнителю.

### Поднимание парты

Задача предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей
- Однотипные работы.
- Исполнители – разнотипные (у каждого своя система команд).
- Согласование по времени.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Играют две бригады по два человека.

Оборудование: стол (парта), который ученики будут поднимать; стол (парта), на который будут опираться первый стол; две каточки с системами команд.

Суть игры: две бригады поднимают стол (парту) и ставят на другой стол (парту).

В системе команд каждой бригады три команды: «Ай», «Ой», «Бй». Первые две команды имеют для разных бригад противоположный смысл. Для одной бригады «Ай» означает «Давить на стол вниз», «Ой» – «Поднимать стол вверх». Для другой – наоборот. Команда «Бй» в обоих случаях означает «поднимать стол вверх». Бригады должны ознакомиться со своими системами команд тайком друг от друга, чтобы каждая бригада знала только свою СКИ.

Ведущий (учитель) поочередно несколько раз отдает команды «Ай» и «Ой». Порядок – не важен. Повторить надо несколько раз. Каждый раз одна команда давит на свой край стола вниз, а другая пытается поднять свой край вверх. Поднять стол при этом не удастся. После этого ведущий дает команды «Бй» и стол легко поднимается.

Обсуждение: Почему в первые разы поднять стол не удалось, а в последний – удалось? В чем причина? Ответ: причина в том, что в последнем случае обе бригады стали поднимать **ОДНОВРЕМЕННО**. То есть действия бригад были согласованы по времени.

### Построение «светофора»

Задача предназначена для освоения следующих элементов параллельного программирования:

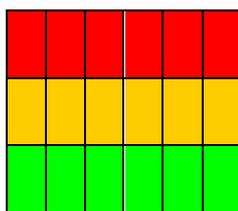
- Совместная работа нескольких исполнителей.
- Однотипные работы.
- Однотипные исполнители.
- Согласование по результатам.
- Соотношение «исполнители–работы» –  $N : 1$ .
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Играют трое.

Оборудование: стол, кубики с гранями трех цветов (красный, желтый, зеленый). Количество кубиков должно быть кратно трем. Лучше, если их будет побольше (15-18). Из этих кубиков будет строиться «светофор».

Суть игры: надо сложить из кубиков «светофор» из трех рядов одинаковой длины: снизу – зеленый, в середине – желтый, сверху – красный.

(рис. 14)



*Рис. 14. Пример светофора из кубиков для игры - Построение «светофора»*

Игроки распределяются по цветам. Каждый из них должен сначала сложить из кубиков ряд своего цвета, а потом установить его в «светофор».

Обсуждение: Ряды кубиков будут, скорее всего, сложены всеми игроками примерно за одно и то же время. Но поле этого «желтому» игроку придется ждать, пока свой ряд уложит «зеленый», а «красному» – пока будут

уложены оба нижних ряда. (Большая длина ряда делает время ожидания более заметным.)

Здесь – отличие как от деления работы на части (приложение 6), так и от согласования по времени (приложение 1-5). В первой игре каждый из землекопов «копал» свою часть канавы независимо от других. Его совершенно не интересовало, какие другие участки уже выкопаны, какие – нет.

Во второй игре требовалась синхронность действий двух бригад.

В последнем случае каждого исполнителя интересует, какую часть работы проделали его коллеги: невозможно поставить верхний ряд кубиков, пока не поставлен нижний.

Но одновременность при этом не требуется (можем складывать «светофор» со скоростью один ряд в день). Важен только результат. Важно, чтобы предыдущий ряд был уложен раньше, чем начнется укладка следующего ряда.

Замечание: важно, что укладка кубиков в «светофор» выполняется сразу целыми рядами.

### **Вариант игры без кубиков.**

Оборудование: 3 пластиковых стаканчика. Фломастеры трех цветов: красный, желтый, зеленый.

Упражнение заключается в том, что игроки берут стаканчики и фломастеры, каждый рисует на своем стаканчике круг своего цвета, После чего из этих стаканчиков встраивается «светофор».

Обсуждение – как в варианте с кубиками.

**Изготовление браслетов**

Задача предназначена для освоения следующих элементов параллельного программирования:

- Согласование по ресурсам.
- Соотношение «исполнители–работы» – 1 : 1.
- Псевдопараллелизм (разделение между исполнителями одного ресурса).
- Разнотипные работы.
- Однотипные исполнители.

Играют 3 человека.

Оборудование:

- a. Лист бумаги – один.
- b. Ножницы – одни.
- c. Фломастеры (ручки, карандаши) – три.
- d. Клей (степпер) – один.
- e. Стол, на котором лежат все материалы.

Задание: каждый из игроков должен изготовить себе бумажный браслет: вырезать его листа бумаги, раскрасить фломастером (рисунок может быть любым) и склеить.

Чтобы при вырезании одного браслета ножницы были заняты большее время, пусть края браслетов будут фигурными.

Суть игры в том, что выделенных ресурсов недостаточно для одновременной независимой работы всех троих игроков. Поэтому кому-то постоянно придется ожидать, пока другие не закончат некоторые действия и не освободят ресурс. Именно поэтому даются ОДНИ ножницы, ОДИН лист бумаги, ОДИН клей. Нехватку этих ресурсов можно противопоставить фломастерам. Их – три. Поэтому на стадии раскрашивания – в отличие от остальных стадий – все три игрока могут действовать одновременно.

Можно попробовать сыграть с двумя фломастерами. Тогда и этого ресурса будет не доставать.

При обсуждении игры вводится понятие «ресурс» (если дети его еще не знают, в некоторых школах оно может быть введено ранее в курсе экономики или в экономической части курса «Окружающий мир»). Обсуждается вопрос о нехватке ресурсов и вынужденных потерях времени на их ожидание.

В процессе обсуждения игры может возникнуть мысль о том, что изменение порядка действий может привести к сокращению времени выполнения упражнения. Это – так. Но это не есть предмет данной игры. Главным образом потому, что в этом упражнении нет четкой системы команд исполнителя и счетчика времени. Поэтому измерить ускорение, полученное в результате изменения порядка действий, невозможно. Этот аспект будет подробно обсуждаться в игре «Постройка замков» (Приложение 10).

**Построение замков**

Задача предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей.
- Однотипные работы.
- Однотипные исполнители.
- Соотношение «исполнители–работы» –  $N : 1$ .
- Загрузка исполнителей.
- Ускорение работы при увеличении количества исполнителей.
- Непропорциональность ускорения.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Игроки:

I. три бригады строителей: две по три человека и одна из двух человек;

II. три наблюдателя, которые должны отследить правильность действий строительных бригад и срок окончания строительства каждой из бригад;

III. «хранитель времени», который «отбивает» такты работы и считает их. В этой игре необходима синхронизация: за один такт работы каждый исполнитель выполняет ровно одну команду.

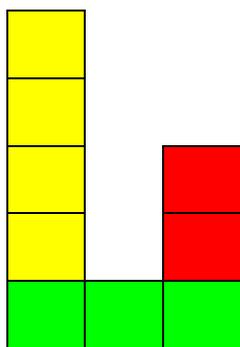
Оборудование:

1) Три отдельных стола (парты) – по одному для каждой строительной бригады. На них будут строиться замки.

2) 27 трехцветных кубиков для строительства замков. Для каждого замка понадобится 3 зеленых кубика, 4 желтых и 2 красных.

3) Карточки или мелкие предметы для отсчета времени. Каждый такт «хранитель времени» будет откладывать в сторону один из предметов и произносить его номер (номер такта работы).

Задача каждой из бригад из девяти кубиков построить замок следующего вида: (рис. 15)



*Рис. 15. Пример постройки из кубиков для игры «Построение замков»*

**Система команд:**

- а. Взять кубик.
- б. Установить кубик. Установить кубик можно либо на стол, либо на кубик, установленный на одном из предыдущих ходов. Невозможно за один ход поставить сразу два кубика. один из которых располагается на другом.
- с. Ничего не делать.

**Три бригады:**

- 1) 3 человека. Каждый строитель ставит кубики только одного цвета: один – только зеленого, второй – только желтого, третий – только красного.
- 2) 3 человека. Все универсалы, т.е. каждый может ставить кубики любого цвета.
- 3) 2 человека. Оба универсалы, т.е. каждый может ставить кубики любого цвета.

Перед началом имеет смысл озвучить следующую логику. Замок состоит из 9 кубиков. Один строитель построит такой замок за 18 ходов. Логично предположить, что двум строителям понадобится  $(18 \cdot 2) = 36$  ходов, трем строителям –  $(18 : 3) = 6$  ходов.

Игру имеет смысл повторить несколько раз. Первый – пробный, для освоения правил. Он вряд ли дойдет до конца. Возможно, пробных понадобится несколько.

Второй раз: сыграть всем вместе для того, чтобы наглядно продемонстрировать разницу во времени завершения строительства разными бригадами. Количество потребовавшихся каждой бригаде ходов записывается.

Третий раз: каждая бригада повторяет отдельно от других. Благодаря этому все могут сосредоточиться на работе одной бригады.

Четвертый раз: после обсуждения полученных результатов. Опять каждая бригада повторяет отдельно от других.

Наилучшие результаты, которые могут быть достигнуты:

I бригада (три специализированных строителя) – 9 ходов,

II бригада (три универсальных строителя) – 8 ходов,

III бригада (два универсальных строителя) – 10 ходов.

Очень может быть, что реальные результаты будут хуже. Особенно полезно, если результат двухчеловечной бригады совпадет с результатом трехчеловечной.

Необходимо обсудить причины увеличения времени строительства по сравнению с теоретическими оценками и разницу в скорости работы двух трехчеловечных бригад.

Разница в скорости работы трехчеловечных бригад объясняется тем, что в первой бригаде строители специализированные. Поэтому строители башен вынуждены пропускать первые ходы, пока для башен не будет готов фундамент.

Обратить внимание на то, что при работе всех бригад обязательно есть хотя бы один простой (пропуск хода). Это простой вызван разной высотой башен. При достраивании более высокой башни на последнем ходе в любом случае работает только один строитель.

Наибольшего времени требует строительство более высокой башни. Значит, надо как можно скорее приступить к строительству именно этой башни. Поэтому первым кубиком в фундаменте должен быть кубик под высокой башней.

Наилучшие результаты будут достигнуты, если строительством башни займутся сразу двое строителей. За один ход один строитель устанавливает на башню кубик, взятый на предыдущем ходу, а второй в это время берет следующий кубик.

## ПРИЛОЖЕНИЕ 11

### Игра «Параллельные процессы: Клинч»

Задача предназначена для освоения следующих элементов параллельного программирования:

- Взаимная блокировка параллельных процессов.
- Соотношение «исполнители–работы» –  $N : N$ .
- Разнотипные работы.
- Однотипные исполнители.
- Псевдопараллелизм.

На столе стоят 2 бутылка с водой (№1 и №2), кружка и кастрюля.

Участники делятся на 3-4 бригады.

Каждая бригада должна составить алгоритм для следующего процесса:

Подойти к столу. Из бутылки перелить в кастрюлю 2 кружки воды из бутылки №1 и 2 кружки из бутылки №2. Вернуться на место.

#### **Система команд:**

1. Подойти к столу
2. Вернуться на место.
3. Взять со стола бутылку №1.
4. Взять со стола бутылку №2.
5. Поставить бутылку на стол.
6. Взять кружку со стола.
7. Поставить кружку на стол.
8. Открыть крышку бутылки.
9. Закрыть крышку бутылки.
10. Взять крышку со стола.
11. Положить крышку на стол.
12. Налить воду из бутылки в пустую кружку.
13. Вылить воду из кружки в кастрюлю.

Подразумевается, что у Исполнителя две руки. То есть он может одновременно держать только два предмета.

Бригада записывает алгоритм на листе флип-чарта.

Все листы с алгоритмами вывешиваются рядом.

Каждая бригада выставляет Исполнителя и Командера (указателя команд). Командер будет зачитывать команды из алгоритма, Исполнитель выполнять их.

Командеры встают около своих листов с алгоритмами.

Определяется порядок, в котором будут работать команды (по номеру команды, по жребию).

Определяется количество тактов работы, приходящихся на одну бригаду. Проще, если оно одинаково для всех. Учитывая длину программ, квант в 4 шага представляется разумным.

Существует Диспетчер, который руководит всеми Командерами. При отдаче команд Командерам Диспетчер отсчитывает количество команд, отданных данной бригаде.

По команде Диспетчера первый в очереди Командер читает очередную команду своего алгоритма. Если Исполнитель в состоянии выполнить команду, он ее выполняет. Командер помечает команду как выполненную. Если Исполнитель не в состоянии выполнить команду, Командер переходит в конец очереди.

Если бригада выполнила запланированное количество команд, она переходит в конец очереди. Начинает работать следующая бригада.

Смысл игры: продемонстрировать конкуренцию процессов за ресурсы.

Интересно, что тупик (клинч, взаимная блокировка процессов) не являются обязательным состоянием в игре.

Ради этого в систему команд включены некоторые уточнения. Например, требование наливать воду именно в пустую кружку, нумерация бутылок (если рядом со столом стоит еще какая-то бутылка, она не может быть задействована, поскольку не имеет номера).

Хорошо дождаться ситуации, когда все бригады заблокируют друг друга и прокрутить несколько циклов, когда ни одна бригада не сможет выполнить свою команду.

Формально слабым местом игры м.б. тот факт, что моделируется выполнение нескольких процессов на одном процессоре, в то время, как современные ЭВМ – многоядерные. Но это слабость чисто формальная. В реале количество процессов может многократно превышать количество ядер. (Чтобы убедиться в этом, достаточно заглянуть в диспетчер программ.) Поэтому ситуация, когда несколько процессов выполняются на одном процессоре – вещь совершенно обыденная. Можно позволить нескольким бригадам действовать одновременно. Это только запутает картину, сделает ее более смазанной.

Можно экспериментировать с количеством ресурсов. Взять 2 кружки и одну бутылку, 2 бутылки и 1 кружку, 2 бутылки и 2 кружки. Заметим, что наличие двух бутылок дает бригадам некоторую свободу. Вдруг кто-то начнет не с бутылки №1, а с бутылки №2.

Можно экспериментировать с дисциплиной обслуживания:

1. Все процессы имеют одинаковый приоритет. Период (количество тактов) постоянный. Величина периода определяется кубиком.
2. Для каждого процесса разыгрывается его приоритет и время активации. Процесс со старшим приоритетом встает в очередь первым.
3. Процесс со старшим приоритетом прерывает работу процесса с младшим приоритетом.

**Игровая обучающая компьютерная программа «Танковый экипаж»**

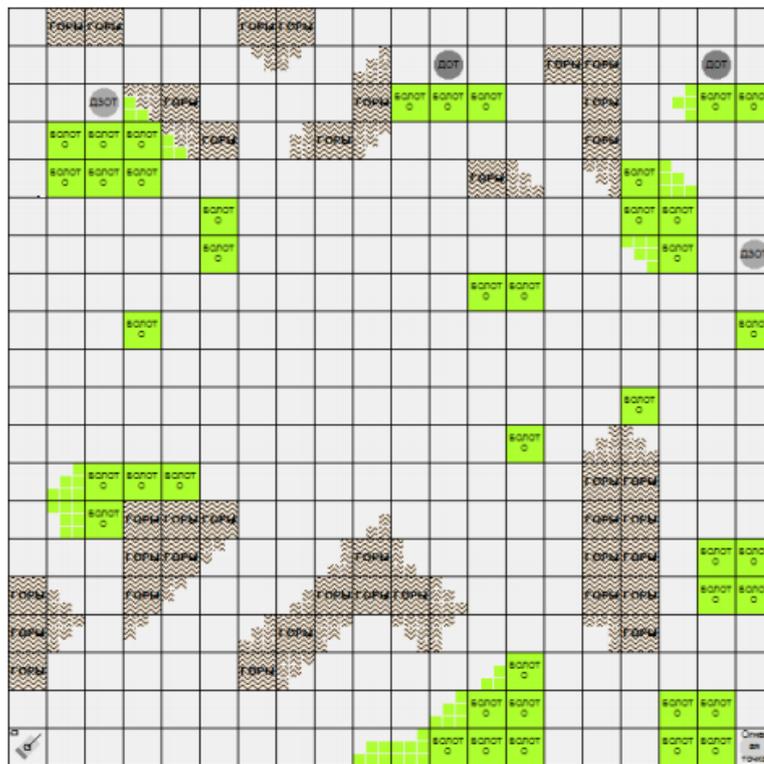
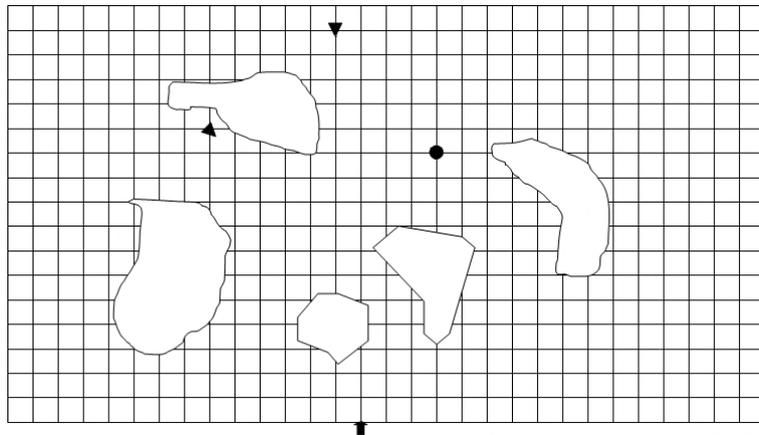
Задача предназначена для освоения следующих элементов параллельного программирования:

- Совместная работа нескольких исполнителей.
- Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).
- Исполнители – разнотипные (у каждого своя система команд).
- Соотношение «исполнители–работы» – N : 1 (весь экипаж в целом решает одну боевую задачу).
- Согласование деятельности исполнителей. Виды согласования: по времени и по результатам (для выстрела танк должен занять нужную позицию, орудие должно быть заряжено; в задание специально включена ситуации, когда командиру и водителю приходится ждать перезарядки орудия).
- Оптимальный порядок действий.

Действия игры происходят на прямоугольной сетке, на которой расположены различные объекты (скалы, болота, дороги, вражеские огневые точки). Танк движется от одного узла сетки к другому с определенной скоростью и может вести огонь из орудия на определенное расстояние. Препятствия – скалы и болота – отличаются тем, что через болото танк может вести огонь, а через скалу – нет. Огневые точки разных типов требуют для уничтожения разного количества попаданий.

Экипаж танка состоит из трех человек: командира-наводчика, водителя и заряжающего. У каждого члена экипажа своя система команд. Наводчик меняет направление орудия и производит выстрел. Водитель управляет движением танка. Заряжающий занимается заряданием орудия.

Ученик должен написать для экипажа танка программу совместных действий, в результате которых будут уничтожены вражеские огневые точки.



*Рис. 16. Пример поля боя для игры «Танковый экипаж»*

На рис. 16 показан пример поля боя. Белым цветом обозначены скалы. Начальное положение и направление танка обозначено стрелкой вниз. Цели, которые надо уничтожить, это:

- Два орудия на открытых позициях. Они обозначены треугольниками. Для их уничтожения требуется одно попадание снаряда;
- ДОТ, обозначен кружком. Требуется двух попаданий.

Дальность стрельбы из орудия – 8 клеток (по горизонтали, вертикали или диагонали). Направление орудия первоначально совпадает с направлением танка. Орудие закреплено на танке. Поэтому при повороте танка на тот же угол поворачивается и орудие. Направление как танка, так и орудия меняется с шагом в 45°. Скорость движения – одна клетка за ход. Для перезарядки орудия надо два хода (Заряжающий выполняет команды Заряжай-1, Заряжай-2).

Возможны ситуации, когда результат совместных действий будет зависеть от того, в каком порядке выполняются действия разных исполнителей на одном ходе. Считаем, что сначала выполняются действия Наводчика, а потом водителя и заряжающего. То есть сначала производится выстрел (с начальной точки движения), потом – перемещение в конечную точку. Заряжать орудие можно начинать на том же ходу, когда произведен выстрел. Но стрелять можно только на следующем ходу, после окончания заряжения (т.е после команды «Заряжай-2»).

Выполнить боевое задание экипажу танка поможет следующая программа (Таблица 7):

*Таблица 7. Пример программного решения для обучающей компьютерной программы «Танковый экипаж»*

<b>№ хода</b>	<b>Наводчик</b>	<b>Водитель</b>	<b>Заряжающий</b>
1.	Пауза	Вправо-вперед	Заряжай-1
2.	Пауза	Влево-вперед	Заряжай-2
3.	Пауза	Вперед	Пауза
4.	Пауза	Вперед	Пауза
5.	Влево 45	Вперед	Пауза
6.	Выстрел	Влево-вперед	Заряжай-1
7.	Вправо 135	Вперед	Заряжай-2
8.	Выстрел	Стоп	Заряжай-1
9.	Пауза	Стоп	Заряжай-2
10.	Выстрел	Вправо-вперед	Заряжай-1
11.	Влево 90	Стоп	Заряжай-2
12.	Выстрел	Стоп	Пауза

Игра представляется достаточно сложной. Но ее сложность хорошо регулируется сложностью игрового поля. Другой регулятор – замена заданий на написание программ заданиями на их чтение (Что произойдет в результате выполнения следующей программы?).

**Сетевой график**

Задача предназначена для освоения следующих элементов параллельного программирования:

1. Совместная работа нескольких исполнителей.
2. Выполнение одной и той же работы одним исполнителем и группой исполнителей. Зависимость скорости выполнения работы от количества исполнителей.
3. Однотипные исполнители (все бригады совершенно равноправны).
4. Однотипные работы.
5. Соотношение «исполнители–работы» –  $N : 1$ .
6. Оптимальный порядок действий. Оптимальная загрузка исполнителей.
7. Согласование деятельности исполнителей по времени и по результатам.
8. Истинный параллелизм (несколько исполнителей одновременно выполняют каждый свои действия).

Дан сетевой график (Рис. 17). Требуется:

1. Изобразить (схематически) сооружение, которое будет построено.
2. Определить, сколько дней потребуется для строительства, если работать будет ровно одна бригада.
3. Определить, сколько дней потребуется для строительства, если работать будет две бригады.
4. Определить, сколько дней потребуется для строительства, если работать будет четыре бригады.
5. Изобразить (схематически) как будет выглядеть стройка через 10 дней после начала строительства, если работать будут 4 бригады.

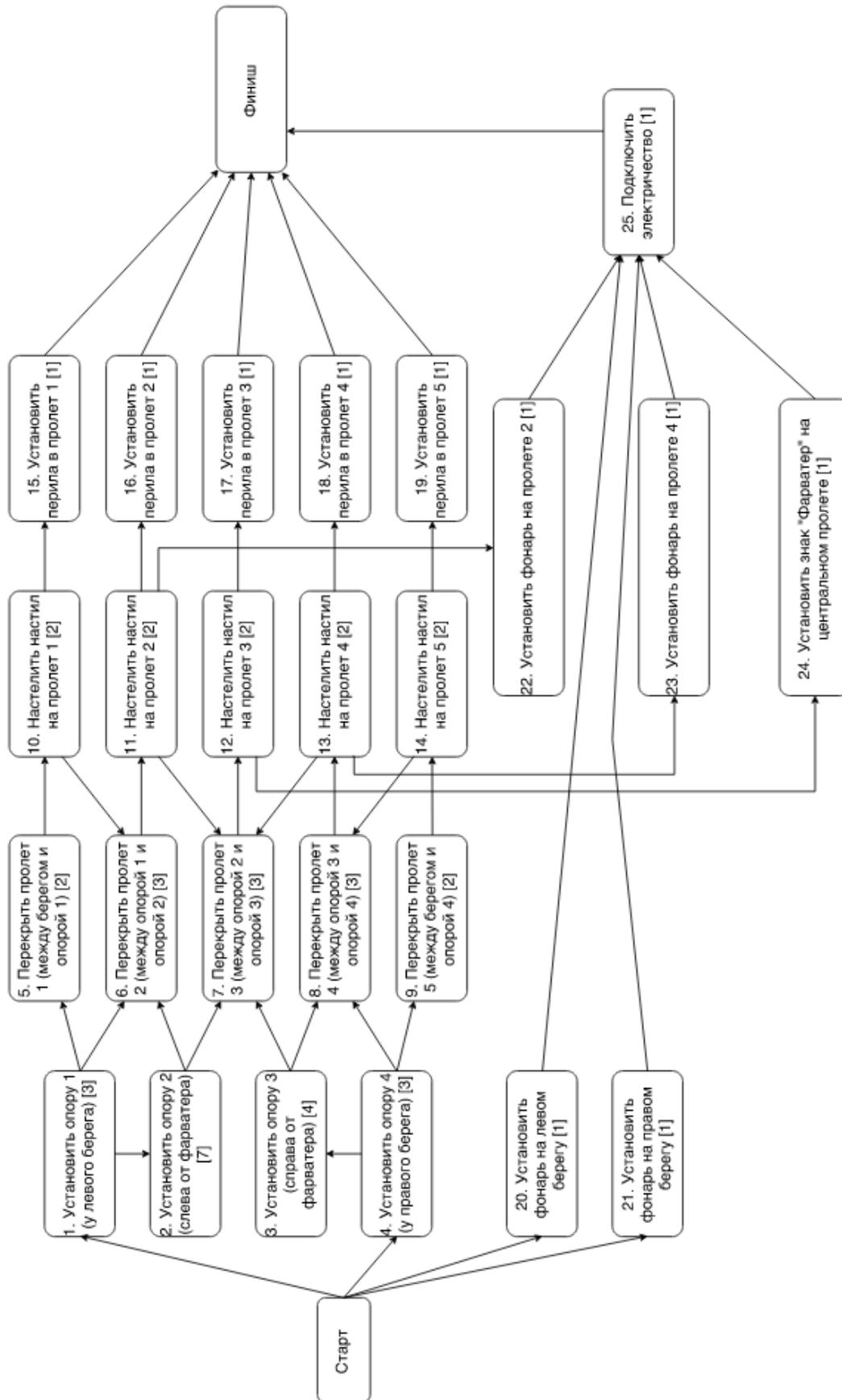


Рис. 17. Пример сетевого графика

Дипломы и сертификаты



**Центр развития научного сотрудничества**

**СЕРПИФКАТ**

участника XI международной молодежной научно-практической конференции

**Шишкин Владислав Андреевич**

принял(а) участие в XI международной молодежной научно-практической конференции

**«Научные исследования и разработки молодых ученых»**

с докладом

**«ВКЛЮЧЕНИЕ ОСНОВ ПАРАЛЛЕЛЬНЫХ  
ВЫЧИСЛЕНИЙ В ШКОЛЬНЫЙ КУРС  
ИНФОРМАТИКИ»**

Россия, г. Новосибирск

Председатель оргкомитета конференции,  
Руководитель Центра развития научного сотрудничества

10 июня 2016 г.

С.С. Чернов





Центр развития научного сотрудничества  
(Новосибирск)

ООО «Центр развития научного сотрудничества»  
ИНН 5402563348 КПП 540201001  
Р/с № 40702810444050099639  
Сибирский банк Сбербанка России  
БИК 045004641  
К/с № 30101810500000000641  
ИНН/КПП 7707083893/540645005

630111, Новосибирск, ул. Кропоткина, 128/3, оф. 18  
Телефон: 8-(383)-291-79-01,  
8-913-749-05-30

Сайт: [www.zms.ru](http://www.zms.ru)  
E-mail: [monography@ngs.ru](mailto:monography@ngs.ru),  
[monography@mail.ru](mailto:monography@mail.ru)

от 3.06.2016 № 55-450

Красноярский государственный педагогический  
университет им. В.П.Астафьева

Шишкину В. А.

Уведомляем о том, что доклад «*ВКЛЮЧЕНИЕ ОСНОВ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В ШКОЛЬНЫЙ КУРС ИНФОРМАТИКИ*», подготовленный и представленный Вами для участия в XI международной научно-практической конференции «Научные исследования и разработки молодых ученых» (Новосибирск: Издательство НГТУ, 2016), рассмотрен оргкомитетом конференции и принят к публикации.

Сборник зарегистрирован и индексируется в наукометрической базе РИНЦ (<http://elibrary.ru/>).

Рассылка сборников материалов конференции планируется не позднее 25 июня 2016 г.

Главный редактор,  
канд. экон. наук, доцент



С.С. Чернов



Центр развития научного сотрудничества  
(Новосибирск)

ООО «Центр развития научного сотрудничества»  
ИНН 5402563348 КПП 540201001  
Р/с № 40702810444050099639  
Сибирский банк Сбербанка России  
БИК 045004641  
К/с № 30101810500000000641  
ИНН/КПП 7707083893/540645005

630111, Новосибирск, ул. Кропоткина, 128/3, оф. 18  
Телефон: 8-(383)-291-79-01,  
8-913-749-05-30

Сайт: [www.zms.ru](http://www.zms.ru)  
E-mail: [monography@ngs.ru](mailto:monography@ngs.ru),  
[monography@mail.ru](mailto:monography@mail.ru)

от 3.06.2016 № 55-461

Красноярский государственный  
педагогический университет  
им. В.П. Астафьева

Шишкину В. А.

Уведомляем о том, что доклад «*ФОРМИРОВАНИЕ ПРЕДСТАВЛЕНИЙ О ПАРАЛЛЕЛЬНЫХ ПРОЦЕССАХ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ*», подготовленный и представленный Вами для участия в XXXIII международной научно-практической конференции «Интеллектуальный потенциал XXI века: ступени познания» (Новосибирск: Издательство ЦРНС, 2016), рассмотрен оргкомитетом конференции и принят к публикации.

Сборник зарегистрирован и индексируется в наукометрической базе РИНЦ (<http://elibrary.ru/>).

Рассылка сборников материалов конференции планируется не позднее 10 июня 2016 г.

Главный редактор,  
канд. экон. наук, доцент



С.С. Чернов

Международный научный центр "Олимп"  
г. Москва, ул. Героев Панфиловцев 17-2, тел. 8 (964) 574-29-46  
conf@olimpiks.ru, www.olimpiks.ru

# СЕРТИФИКАТ

выдан

**ШИШКИНУ В.А.**

за участие в международном конкурсе:

**«Лучшая научно-исследовательская работа-2016»**

Руководитель НЦ "Олимп"  
**Палаткина С.В.**



г. Москва, июнь 2016г.

Институт управления и социально-экономического развития

# СЕРТИФИКАТ

за публикацию в е-журнале  
«Теория и практика современной науки»

автор(ы): *Шишкин В.А.*

**ФОРМИРОВАНИЕ ПРЕДСТАВЛЕНИЙ О ПАРАЛЛЕЛЬНЫХ ПРОЦЕССАХ В  
ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ**



Ген. директор ООО «ИУСЭР»  
Чернышова О.А.

дата выдачи сертификата:  
06.06.2016

**ООО «Институт управления и социально-экономического развития»**

ИНН 6454110943, КПП 645001001, р/с № 40702810000030005711  
Банк: Филиал "Саратовский" ПАО "Ханты-Мансийский банк Открытие"  
БИК 046311900, к/с № 30101810663110000900  
Адрес: г.Саратов, ул.Волжская, д.28, 2/10

[iuser.saratov@mail.ru](mailto:iuser.saratov@mail.ru); [www.iupr.ru](http://www.iupr.ru); 8 (917) 021 49 78

**СПРАВКА**

Подтверждаю, что статья *Шшикина В.А.* «ФОРМИРОВАНИЕ ПРЕДСТАВЛЕНИЙ О ПАРАЛЛЕЛЬНЫХ ПРОЦЕССАХ В ШКОЛЬНОМ КУРСЕ ИНФОРМАТИКИ» принята к публикации в е-журнале «Теория и практика современной науки» № 6(12)2016 ISSN 2412-9682

Свидетельство о регистрации средства массовой коммуникации Эл № 61970 от 02.06.2015г. Издание зарегистрировано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникации (Роскомнадзор)

Выпуски журнала включены в РИНЦ (e-library), договор №435-06/2015

Ген. директор  
Чернышова О.А.

