



КОМПЕТЕНТНОСТНЫЙ ПОДХОД В ОБУЧЕНИИ УЧАЩИХСЯ ПРОГРАММИРОВАНИЮ (с приложением)

А.И. Лапо

Лицей Белорусского государственного университета, Беларусь

Аннотация. В данной статье рассмотрено применение компетентностного подхода к обучению программированию в рамках учебного предмета «Информатика». Выделены структурные элементы алгоритмической компетентности, которые формируются на различных этапах обучения. Поэтапное формирование алгоритмической компетенции позволит повысить уровень информационно-коммуникационной компетентности учащихся и будет способствовать их профориентации при выборе профессий ИКТ-сферы. В приложении приводится система задач (задачный комплекс) по теме «Команды ветвления», VIII класс.

Ключевые слова: алгоритмическая компетентность, программирование, системно-задачный подход, школьная информатика.

COMPETENCY-BASED APPROACH IN TEACHING STUDENTS PROGRAMMING

A. I. Lapo

Lyceum of Belarusian State University, Belarus

Abstract. The article describes the use of competence approach to teaching programming as part of a school course of computer science. The structural elements of algorithmic competence, which are formed at different stages of training, are highlighted. The gradual formation of algorithmic competence will improve the information and communication competence of the student, and will contribute to its choice of career counseling professions pi ICT sector. The annex provides a system of tasks (task complex) on the topic "Branching Commands", VIII class.

Keywords: algorithmic competence, programming, system-task approach, school course of computer science.

Введение

Компетентностный подход в образовании – явление достаточно новое и не до конца изученное. В Беларуси переход на компетентностный подход в школьном образовании начался в 2015 году, когда стал вопрос о необходимости пересмотра вопросов содержания и организации образования.

В России говорить о компетентностном подходе в образовании начали в начале 2000-х. В своем докладе "Определение общепредметного содержания и ключевых компетенций как характеристика нового подхода к конструированию образовательных стандартов" на Отделении философии образования и теоретической педагогики Российской академии образования д.п.н. А.В. Хуторской говорит о необходимости включения ключевых компетенций в структуру разрабатываемого стандарта. Им же были предложены и сами ключевые компетенции [1]:

- ✓ *Ценностно-смысловые компетенции.*
- ✓ *Общекультурные компетенции.*
- ✓ *Учебно-познавательные компетенции.*
- ✓ *Информационные компетенции.*
- ✓ *Коммуникативные компетенции.*
- ✓ *Социально-трудовые компетенции.*
- ✓ *Компетенции личностного самосовершенствования.*

В дальнейшем А.В. Хуторским была разработана технология проектирования компетентностей и определены уровни иерархии компетентностей: метапредметные (или ключевые), общепредметные и предметные [2].

И.А. Зимняя [3], анализируя развитие компетентностного подхода, выделяет три группы компетенций:

- ✓ *Компетенции, относящиеся к самому человеку как личности, субъекту деятельности, общения.*
- ✓ *Компетенции, относящиеся к социальному взаимодействию человека и социальной сферы.*
- ✓ *Компетенции, относящиеся к деятельности человека.*

В последнюю группу компетенций И.А. Зимняя включает наряду с компетенциями познавательной деятельности также компетенции информационных технологий.

Компетентностному подходу в изучении информатики посвящена статья А.П. Шестакова [4]. В качестве ключевых компетенций рассматриваются

интеллектуальная, организационная, информационная. Автором сформулированы и предметные компетенции:

- ✓ *алгоритмическая;*
- ✓ *вычислительная;*
- ✓ *наглядно-модельная;*
- ✓ *прогностическая;*
- ✓ *исследовательская;*
- ✓ *методологическая.*

Различные подходы к определению компетенций и построению их иерархии приводят к тому, что некоторые компетенции могут быть рассмотрены как на метапредметном, так и на частнопредметном уровне.

Формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ–компетенции) может быть отнесено к метапредметным результатам (таким образом ИКТ-компетенции прописаны в российских Федеральных государственных образовательных стандартах (ФГОС)). С другой стороны, содержательно ИКТ относятся к предметной области «Информатика», следовательно, ИКТ – компетенции могут рассматриваться как предметные компетенции.

В.А. Адольф и И.Ю. Степанова [5] рассматривают информационную компетенцию на трех уровнях иерархии: как ключевую, базовую (предметную) и специальную (для решения конкретной проблемы). Там же выделены уровни проявления информационной компетентности:

- элементарный (репродуктивная деятельность);
- функциональный (репродуктивная деятельность с элементами творческой деятельности);
- системный (индивидуально-творческая деятельность).

И.Н. Елисеев [6, с.81] говорит об интегративной сути компетенции «поскольку ее наличие определяется не объемом усвоенной информации, а системой освоенных и опробованных в действии методов поиска недостающих знаний на основе интеграции имеющихся». Свойство интегративности позволяет анализировать одну и ту же компетенцию как на мета уровне, так и на более низких уровнях: общепредметном, предметном или специальном.

В данной статье алгоритмическая компетенция будет рассмотрена на предметном уровне.

Алгоритмическая компетентность

В литературе встречаются различные описания алгоритмической компетентности:

- владение алгоритмами (А.П. Шестаков [4]);
- «Теоретические модели + Алгоритмы = Алгоритмическая компетентность» (М.В. Кондурар [7]);
- моделирование и структурирование знаний, конструирование алгоритмов, анализ алгоритмов (Лобанова Н.В., Маньшин М.Е., Смыковская Т.К [8]).

М.Е. Маньшин и А.А. Бабенко [9] определяют алгоритмическую компетентность как «системное и динамическое образование, характеризующееся определенным уровнем развития алгоритмического мышления, осознанием общих компонентов алгоритмизации и проявляющееся в разнообразных формах алгоритмической деятельности, побуждаемой потребностно-мотивационной сферой».

О.Н. Ярыгин ([10, с.36]) утверждает, что «алгоритмическая компетентность состоит не только в способности конструирования, описания и выполнения алгоритмов, но и в определении свойств алгоритмов, распространении их на другие задачи, в алгоритмизации неформализованных проблем». Он выделяет составные компоненты алгоритмической компетентности [11, с. 14]:

- абстрагирование, моделирование и структурирование знаний;
- конструирование (синтез) алгоритмов;
- анализ алгоритмов;
- алгоритмическая нотация.

Для описания структуры компетентности О.Н. Ярыгин вводит «тетраэдральную» модель, с четырьмя гранями: **алгоритмическая, языковая, дедуктивная** (логическая) и **индуктивная** (рис. 1).

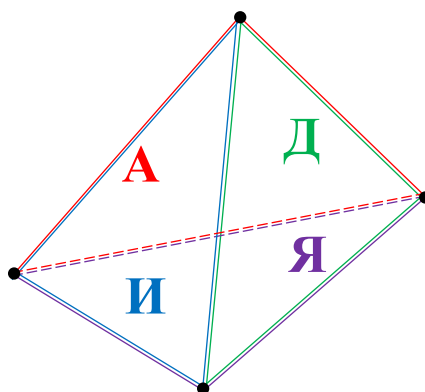


Рисунок 1 – Тетраэдральная модель компетентности (О.Н. Ярыгин)

Компетентность проявляется через качества личности, представляет собой уровень сформированности соответствующей *компетенции*, проявляется через мотивы и ценностные установки личности в процессе реализации компетенции.

О.В. Шемет [12] показывает, что любая компетенция складывается из трех основных компонентов: когнитивного, интегративно-деятельностного и личностного.

Когнитивный компонент связан со знаниями и способами их получения. О.Н. Ярыгин разделяет знания на **декларативные** («знаю, что») и **процедурные** («знаю, как»). В рамках «тетраэдральной» модели им соответствуют языковая и алгоритмическая грани. Интегративно-деятельностному компоненту соответствует логическая (дедуктивная) грань. Индуктивная грань соответствует личностному компоненту, поскольку обеспечивает рефлексию, саморазвитие, получение неявных знаний, порождаемых опытом, интуицией и воображением. Таким образом различные структурные модели компетентности можно свести друг к другу.

В таблице 1 представлена алгоритмическая компетентность учащегося общеобразовательного учебного заведения на предметном уровне в проекции различных моделей компетентности.

Таблица 1. Сравнение различных моделей компетентности.

«Тетраэдральная» модель (О.Н. Ярыгин)	Модель компетентности (О.А. Шемет)	Алгоритмическая компетентность учащегося на предметном уровне
Алгоритмическая грань	Когнитивный компонент («знаю, что»)	Алгоритмические конструкции (следование, ветвление, цикл, пошаговая детализация) и различные типы данных
Языковая грань	Когнитивный компонент («знаю, как»)	Основы языка программирования
Дедуктивная грань	Интегративно-деятельностный компонент	Формализация и моделирование
Индуктивная грань	Личностный компонент	Тестирование, отладка, анализ, исследование решенной задачи

Каждая из компонентов алгоритмической компетентности учащегося, в свою очередь подразделяется на более мелкие структурные единицы, в зависимости от уровня изучения темы «Основы алгоритмизации и программирования» и возрастных особенностей учащихся. Структурные элементы алгоритмической компетентности рассмотрены в [14]. Анализ нормативных документов (Образовательные стандарты общего среднего образования, Нормы оценки результатов учебной деятельности учащихся по учебному предмету «Информатика», Учебная программа по учебному предмету «Информатика») системы образования Республики Беларусь позволяют

выделить отдельные компетенции, формирование и развитие которых приводит к сформированности алгоритмической компетентности. Выделенные компетенции сгруппированы в соответствии с последовательными этапами изучения информатики в общеобразовательных учебных заведениях. Такое деление позволит учителю планировать формирование и развитие алгоритмической компетентности учащихся.

Пропедевтический этап (VI–VII класс).

- АК1. Алгоритмические компетенции, связанные с выполнением готового алгоритма;
- АК2. Алгоритмические компетенции, относящиеся к изменению алгоритма при изменении обстановки исполнителя;
- АК3. Алгоритмические компетенции, обеспечивающие запись алгоритма с помощью системы команд исполнителя;
- АК4. Алгоритмические компетенции, касающиеся поиска и исправления ошибок в алгоритме;
- АК5. Алгоритмические компетенции разработки и выполнения простейших алгоритмов с использованием цикла, ветвления, вспомогательного алгоритма.

Базовый этап (VIII–IX класс).

- АК6. Алгоритмические компетенции в области формализации задачи;
- АК7. Алгоритмические компетенции, позволяющие осуществлять выбор типов данных для реализации алгоритма;
- АК8. Алгоритмические компетенции, обеспечивающие запись алгоритма на языке программирования;
- АК9. Алгоритмические компетенции в области отладки программы;
- АК10. Алгоритмические компетенции разработки алгоритмов, записи и выполнения программ для обработки числовых и символьных данных с использованием цикла, ветвления, вспомогательного алгоритма.

Общеобразовательный этап (X–XI класс).

- АК11. Алгоритмические компетенции в области хранения данных в памяти компьютера;
- АК12. Алгоритмические компетенции моделирования учебных задач с использованием структурированных типов данных;
- АК13. Алгоритмические компетенции, осуществляющие применение разработанных алгоритмов для решения других задач;

АК14. Алгоритмические компетенции, относящиеся к использованию элементов управления для разработки оконных интерфейсов программ.

Профильный этап (X–XI класс).

АК15. Алгоритмические компетенции в области разработки алгоритмов и программ с применением знаний теории алгоритмов и использованием различных структур данных;

АК16. Алгоритмические компетенции в области исследовательской деятельности с применением знаний, умений и навыков алгоритмизации и программирования;

АК17. Алгоритмические компетенции в области проектной деятельности, проявляющиеся в самостоятельной разработке готовых программных продуктов.

Указанные компетенции базового уровня (АК6 – АК10) развиваются на основе приведенных выше компетенций пропедевтического уровня (АК1 – АК5). Так, алгоритмическая компетенция АК8 развивается на основе компетенции АК3, компетенция АК9 – на основе АК4, компетенция АК10 – на основе АК5. Компетенции АК1, АК2 переходят к компетенции АК10 и «доформируются» знаниями и умениями, относящимися к конкретному языку программирования.

Формирование компетенций происходит таким образом, что компетенции АК11 и АК12 совершенствуются на основе компетенции АК7, компетенции АК13 и АК14 развиваются с учетом компетенций АК8 – АК10. Компетенция АК6 становится составной частью АК13. Состав алгоритмических компетенций АК1 – АК14 отвечает перечню требований, предъявляемых к выпускнику учреждений общего среднего образования в области алгоритмизации и программирования.

Компетенции АК15 – АК17 развиваются на основе компетенций АК1 – АК14. Профильный этап предполагает осознанный выбор учащимся предмета для дальнейшего изучения, следовательно, на этом этапе у учащихся повышенная мотивация к учебной предметной деятельности. Через мотивы (и рефлексию) происходит проявление личностной компоненты алгоритмической компетентности. Компетенция АК15 относится к деятельностному компоненту алгоритмической составляющей. Компетенции АК16 – АК17 соответствуют личностному компоненту алгоритмической компетентности.

Таким образом, все формируемые компетенции связаны между собой, лежащие в их основе умения и навыки развиваются на основе учебной деятельности, личностные качества – на основе психологических законов развития личности. Взаимосвязь

компетенций, формируемых в процессе изучения алгоритмизации и программирования в учреждениях общего среднего образования приведена на рис. 2. Различные цвета на рисунке показывают: какую составляющую алгоритмической компетентности будет формировать соответствующая компетенция: синий цвет – когнитивная, зеленый – деятельностная (бирюзовый – репродуктивная, ярко-зеленый – продуктивная, салатный – творческая), желтый – личностная.

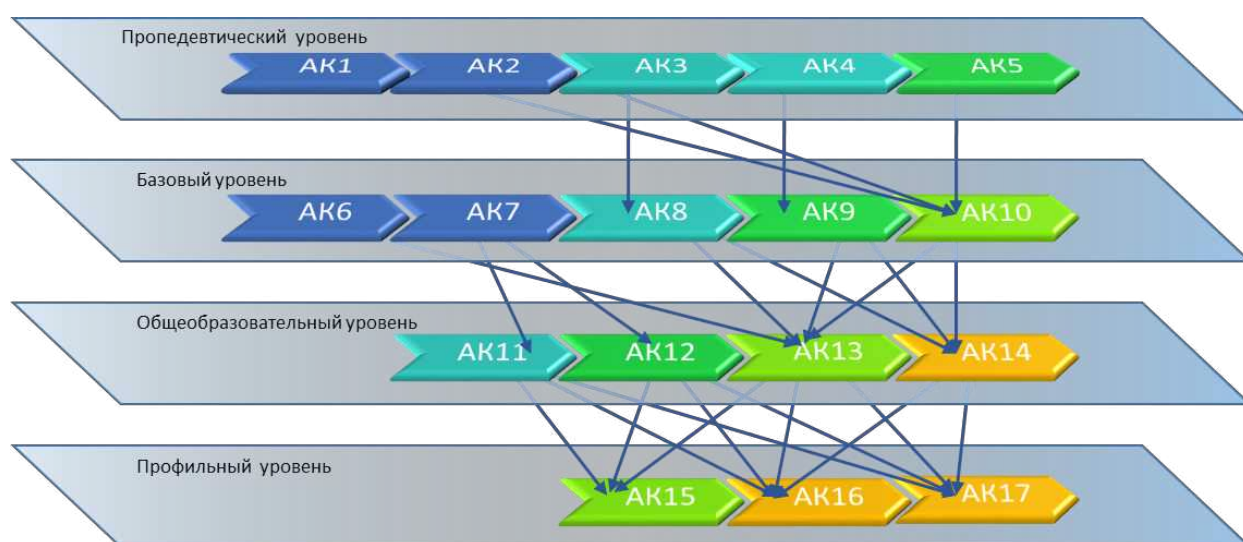


Рисунок 2 – Алгоритмические компетенции на различных уровнях обучения информатике

Осваиваемые элементы алгоритмической компетентности разделены по классам весьма условно. Поскольку компетентность проявляется как свойство личности, то для различных учащихся осваиваемые элементы могут формироваться как раньше, так и позже выделенных этапов. Возможно взаимопроникновение на разные уровни как формируемых элементов компетенции, так и элементов знаний, связанных с ними. Например, в рамках пропедевтического курса, при работе с исполнителями, может быть введено понятие числовой переменной, которое используется для решения задач повышенной сложности (посчитать количество закрасенных клеток на поле Робота или суммарную длину отрезков, нарисованных Чертежником). Такие задачи будут носить пропедевтический характер не только для дальнейшего знакомства с основными алгоритмическими конструкциями, но и для методов алгоритмизации (задачи поиска, элементы вычислительной геометрии и др.)

Поскольку «тетраэдральная» модель компетентности является полным графом (граф, в котором каждая пара различных вершин смежна), то формирование компетентности не может проходить линейно – все составляющие компетентности должны взаимодействовать друг с другом.

Схематично процесс обучения основам алгоритмизации и программирования учащихся учреждений общего среднего образования при компетентностном подходе можно изобразить следующим образом (рис. 3).



Рисунок 3 – Модель процесса обучения основам алгоритмизации и программирования учащихся учреждений общего среднего образования при компетентностном подходе

Важным элементом предложенной модели являются обратные связи. Особая роль принадлежит рефлексии, которая позволяет осознать наличие проблемы и необходимость поиска средств для ее решения. Компетенции, являясь требованием (нормой) к образовательной подготовке учащегося, реализуются через содержание и цели обучения в процессе учебной деятельности ученика. Наличие обратной связи между деятельностью и знаниями приводит ученика к осознанию необходимости самостоятельного поиска новых знаний (команд языка программирования) для реализации практической деятельности.

Системно-задачный подход

В.А. Адольф и И.Ю. Степанова [5] основывают формирование информационной компетенции на одной из триад компетентностного подхода: «знания – задача – деятельность». Именно задача обеспечивает наличие прямой связи между знаниями и деятельностью.

Для реализации компетентного подхода при обучении программированию в рамках учебного предмета «Информатика» может использоваться системно-задачный подход. Здесь для каждой темы разрабатывается система задач. Система задач строится на синергетических принципах. Использование синергетических моделей для обучения учащихся программированию рассматривается в работах С.М. Окулова [14] и [15].

Методика построения систем задач (задачных комплексов) описана нами в работе [16] и включает следующие этапы:

- Определение фундаментальных составляющих системы (С. М. Окулов называет их аттракторами). Они определяются целями, задачами и содержанием учебной программы.
- Выбор родовых задач, которые покрывают фундаментальные составляющие системы. Обучение решению таких задач осуществляется «при помощи подражания и опыта». Это этап линейного развития системы синергетической системы, который готовит скачек – переход к возможности решения индивидуальных задач.
- Разработка индивидуальных задач, соответствующих родовым. Решение индивидуальных задач должно основываться на решении родовых. Но вместе с тем, индивидуальные задачи должны иметь различный уровень сложности, для того, чтобы обеспечить личностное развитие учащегося. Только в этом случае принцип нелинейности обеспечит переход на новый уровень.
- Разработка индивидуальных задач, требующих применения опыта решения родовых задач в измененных или новых условиях.

При построении указанной системы используются задачи разных типов (*примеры задач приведены в приложении*, при этом для индивидуальных задач предложено 12 вариантов, однако, при использовании на уроках, их количество можно увеличить или уменьшить). Приводимая ниже типология задач основана на приемах деятельности, которые будут применяться учащимися для решения задачи:

- А) задача с разобранным алгоритмом и готовой программой.
- В) задача с частично описанным алгоритмом и программой с пропусками.
- С) задача, аналогичная задаче с разобранным алгоритмом и готовой программой.
- Д) задача, требующая применения стандартных подходов, но без разбора аналогичной.
- Е) нестандартная задача, задача повышенного уровня сложности.

Для родовых задач обязательна разработка задач типа А (задача с разобранным алгоритмом и готовой программой).

Задачи типа С (задача, аналогичная задаче с разобранным алгоритмом и готовой программой) и D (задача, требующая применения стандартных подходов, но без разбора аналогичной) разрабатываются как индивидуальные. Количество индивидуальных вариантов зависит от количества учащихся в группе и уровня их персональной подготовки.

Задачи типа В (задача с частично описанным алгоритмом и программой с пропусками) разрабатываются в том случае, если уровень подготовки учащихся низкий или тема достаточно сложная и плохо усваивается учащимися.

Задачи типа Е разрабатываются для учащихся с высоким уровнем подготовки.

Предложенная методика построения систем задач для обучения программированию учащихся учреждений общего среднего образования слабо привязана к конкретному языку программирования. Поскольку тексты программ (полные или с пропусками) встречаются только для задач типов А и В, то при использовании различных языков программирования достаточно заменить лишь их. Этапы решения задачи, которые описаны для задач типов А или В, сохранятся в неизменном виде. В приложении приведены примеры задач с использованием языков Pascal и C++.

Для оценки уровня сформированности компетентности может быть применен следующий подход:

- Когнитивный компонент компетенции оценивается с помощью теста, включающего вопросы различного типа по изученной теме.
- Деятельностный компонент определяется умением решать задачи.
- Личностный компонент проявляется в способности к решению задач в нестандартной (не разобранной на уроке) ситуации или при решении задач повышенного уровня сложности. Критериями сформированности личностного компонента алгоритмической компетентности может служить также участие учащегося в различных олимпиадах, конференциях, предметных конкурсах по информатике. Выпускники учреждений общего среднего образования, со сформированным личностным компонентом алгоритмической компетенции, часто выбирают ИТ-сферу для получения дальнейшего образования.

Педагогический эксперимент, проведенный с учащимися VIII и X классов в пяти различных учебных заведениях показал эффективность системно-задачного подхода

для формирования предметной алгоритмической компетентности. В VIII классе, в рамках темы «Основы алгоритмизации и программирования» учащиеся изучали алгоритмические конструкции ветвления и цикла, а в X классе – символьный и строковый типы данных.

В таблице 2 показаны результаты учащихся до эксперимента и после его окончания в контрольной и экспериментальной группах.

Таблица 2. Оценки учащихся.

	8 класс (41 человек в контрольной группе, 40 – в экспериментальной)	10 класс (44 человека в контрольной группе, 45 – в экспериментальной)																																																																											
До эксперимента	<p>До начала эксперимента 8 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> <tr><th>Контрольная группа</th><td>0</td><td>1</td><td>6</td><td>5</td><td>9</td><td>5</td><td>5</td><td>3</td><td>5</td><td>0</td><td>2</td></tr> <tr><th>Экспериментальная группа</th><td>0</td><td>2</td><td>4</td><td>2</td><td>6</td><td>3</td><td>10</td><td>1</td><td>4</td><td>0</td><td>7</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	Контрольная группа	0	1	6	5	9	5	5	3	5	0	2	Экспериментальная группа	0	2	4	2	6	3	10	1	4	0	7	<p>До начала эксперимента 10 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr><th>Контрольная группа</th><td>4</td><td>10</td><td>1</td><td>4</td><td>4</td><td>2</td><td>2</td><td>7</td><td>7</td><td>3</td><td>0</td><td>0</td></tr> <tr><th>Экспериментальная группа</th><td>7</td><td>5</td><td>6</td><td>5</td><td>1</td><td>1</td><td>6</td><td>5</td><td>7</td><td>0</td><td>2</td><td>0</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	11	Контрольная группа	4	10	1	4	4	2	2	7	7	3	0	0	Экспериментальная группа	7	5	6	5	1	1	6	5	7	0	2	0
Баллы	0	1	2	3	4	5	6	7	8	9	10																																																																		
Контрольная группа	0	1	6	5	9	5	5	3	5	0	2																																																																		
Экспериментальная группа	0	2	4	2	6	3	10	1	4	0	7																																																																		
Баллы	0	1	2	3	4	5	6	7	8	9	10	11																																																																	
Контрольная группа	4	10	1	4	4	2	2	7	7	3	0	0																																																																	
Экспериментальная группа	7	5	6	5	1	1	6	5	7	0	2	0																																																																	
После эксперимента (теория)	<p>После эксперимента (теория) 8 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> <tr><th>Контрольная группа</th><td>2</td><td>1</td><td>1</td><td>5</td><td>10</td><td>7</td><td>8</td><td>7</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>Экспериментальная группа</th><td>2</td><td>0</td><td>0</td><td>0</td><td>6</td><td>9</td><td>7</td><td>4</td><td>7</td><td>0</td><td>0</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	Контрольная группа	2	1	1	5	10	7	8	7	0	0	0	Экспериментальная группа	2	0	0	0	6	9	7	4	7	0	0	<p>После эксперимента (теория) 10 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr><th>Контрольная группа</th><td>0</td><td>2</td><td>1</td><td>2</td><td>8</td><td>6</td><td>4</td><td>9</td><td>8</td><td>2</td><td>2</td><td>0</td></tr> <tr><th>Экспериментальная группа</th><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>6</td><td>1</td><td>10</td><td>14</td><td>8</td><td>4</td><td>0</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	11	Контрольная группа	0	2	1	2	8	6	4	9	8	2	2	0	Экспериментальная группа	0	0	0	1	1	6	1	10	14	8	4	0
Баллы	0	1	2	3	4	5	6	7	8	9	10																																																																		
Контрольная группа	2	1	1	5	10	7	8	7	0	0	0																																																																		
Экспериментальная группа	2	0	0	0	6	9	7	4	7	0	0																																																																		
Баллы	0	1	2	3	4	5	6	7	8	9	10	11																																																																	
Контрольная группа	0	2	1	2	8	6	4	9	8	2	2	0																																																																	
Экспериментальная группа	0	0	0	1	1	6	1	10	14	8	4	0																																																																	
После эксперимента (практика)	<p>После эксперимента (практика) 8 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr> <tr><th>Контрольная группа</th><td>0</td><td>1</td><td>10</td><td>6</td><td>5</td><td>3</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td></tr> <tr><th>Экспериментальная группа</th><td>0</td><td>1</td><td>9</td><td>5</td><td>6</td><td>5</td><td>0</td><td>2</td><td>0</td><td>0</td><td>0</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	Контрольная группа	0	1	10	6	5	3	0	2	0	0	0	Экспериментальная группа	0	1	9	5	6	5	0	2	0	0	0	<p>После эксперимента (практика) 10 класс</p> <table border="1"> <tr><th>Баллы</th><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td></tr> <tr><th>Контрольная группа</th><td>0</td><td>6</td><td>2</td><td>6</td><td>11</td><td>4</td><td>4</td><td>4</td><td>1</td><td>7</td><td>3</td><td>0</td></tr> <tr><th>Экспериментальная группа</th><td>0</td><td>1</td><td>2</td><td>4</td><td>5</td><td>3</td><td>5</td><td>10</td><td>11</td><td>10</td><td>4</td><td>0</td></tr> </table>	Баллы	0	1	2	3	4	5	6	7	8	9	10	11	Контрольная группа	0	6	2	6	11	4	4	4	1	7	3	0	Экспериментальная группа	0	1	2	4	5	3	5	10	11	10	4	0
Баллы	0	1	2	3	4	5	6	7	8	9	10																																																																		
Контрольная группа	0	1	10	6	5	3	0	2	0	0	0																																																																		
Экспериментальная группа	0	1	9	5	6	5	0	2	0	0	0																																																																		
Баллы	0	1	2	3	4	5	6	7	8	9	10	11																																																																	
Контрольная группа	0	6	2	6	11	4	4	4	1	7	3	0																																																																	
Экспериментальная группа	0	1	2	4	5	3	5	10	11	10	4	0																																																																	

В таблице 3 приведены оценки проведенного эксперимента по критерию Вилкоксона- Манна-Уитни с уровнем значимости 0,05.

Таблица 3. Оценки педагогического эксперимента.

8 класс				
До эксперимента		После эксперимента		
Эмпирическое значение	Критическое значение	Эмпирическое значение (теория)	Эмпирическое значение (практика)	Критическое значение
1,488	1,96	3,028	3,972	1,96
10 класс				
До эксперимента		После эксперимента		
Эмпирическое значение	Критическое значение	Эмпирическое значение (теория)	Эмпирическое значение (практика)	Критическое значение
0,451	1,96	2,642	3,438	1,96

Согласно данным полученным до эксперимента можно утверждать, что характеристики экспериментальной и контрольной групп совпадают с уровнем значимости 0.05. После эксперимента достоверность различий характеристик сравниваемых групп составляет 95%. Следовательно, можно сделать вывод, что эффект изменений обусловлен именно применением методики обучения, основанной на системно-задачном подходе.

Проведенный эксперимент позволил отследить динамику изменения когнитивной и деятельностной составляющих алгоритмической составляющей.

Системно-задачный подход используется в Лицее БГУ около 10 лет. Среди выпускников лицея в течение этого времени 8 победителей Международной олимпиады по информатике, 14 победителей Международной Жаутыковской олимпиады, 3 победителя Международной конференции ICYS, более 100 победителей Белорусской республиканской олимпиады. С 2006 года лицеисты являются постоянными участниками Всероссийской командной олимпиады школьников по программированию, где завоевали 6 медалей и 17 дипломов разного достоинства. Ежегодное участие в Открытой олимпиаде по программированию (г. Москва) принесло Лицею более 40 дипломов, из которых 14 – дипломы победителей (1 степень).

Эти факты, а также то, что порядка 70% выпускников физико-математических классов Лицея БГУ выбирают для дальнейшего обучения университеты и факультеты с ориентацией на ИТ-специальности, говорит о положительном влиянии системно-задачного подхода на формирование личностной компоненты алгоритмической компетенции.

Заключение

Положительные результаты педагогического эксперимента по формированию алгоритмической компетентности на основе использования системно-задачного подхода при обучении учащихся учреждений общего среднего образования позволяют рекомендовать его для внедрения в учебный процесс.

Современные учебные пособия, изданные в Беларуси за последние три года [17]–[20], ориентированы на применение системно-задачного подхода при обучении учащихся основам алгоритмизации и программирования.

Также этот подход использован при разработке практико-ориентированных задач, рекомендованных Национальным институтом образования для проверки и оценки качества знаний учащихся при осуществлении текущего и промежуточного контроля в процессе фронтальной, индивидуальной или групповой работы [21], [22].

Список библиографических ссылок (на языке оригинала)

1. Хуторской А. Ключевые компетенции как компонент личностно-ориентированного образования. *Народное образование*. 2003;2:58–64.
2. Хуторской А. Ключевые компетенции. Технология конструирования. *Народное образование*. 2003;5:55–61.
3. Зимняя И.А. Ключевые компетенции – новая парадигма результата современного образования. *Эксперимент и инновация в школе*. 2009;2:7–14.
4. Шестаков А.П. Компетентностный подход в обучении информатике: контрольно-измерительные материалы. *Информатика и образование*. 2010;6:57–65.
5. Адольф В.А., Степанова И.Ю. Методологические подходы к формированию информационной культуры педагога. *Информатика и образование*. 2006;1:2–5.
6. Елисеев И.Н. Методология оценки уровня компетенций студента. *Информатика и образование*. 2012;4:80–85.
7. Кондурар М.В. Развитие алгоритмической компетентности при интегрированном изучении дискретной математики и информатики студентами колледжа. *Вектор науки*. ТГУ. 2014;1:235–238.
8. Лобанова Н.В., Маньшин М.Е., Смыковская Т.К. Структура и формирование интеллектуальной компетентности будущего учителя с дополнительной специальностью «информатика». *Среднее профессиональное образование*. 2010;3:42–46.
9. Маньшин М.Е., Бабенко А.А. Формирование алгоритмической компетентности будущего учителя информатики при изучении баз данных и систем управления базами данных. [Электронный ресурс]. *Грани познания: электронный научно-образовательный журнал*; 2012;6(20). URL: <http://www.grani.vspu.ru/jurnal/25> (дата обращения: 09.12.2019).
10. Ярыгин О.Н. *Методология формирования компетентности в аналитической деятельности при подготовке научных и научно-педагогических кадров*: автореферат диссертации на соискание ученой степени доктора педагогических наук 13.00.08. Тольятти, 2012. 42 с.
11. Ярыгин О.Н. *Формирование интеллектуальной компетентности студентов ИТ-специальностей в процессе изучения дискретной математики*: автореферат диссертации на соискание ученой степени кандидата педагогических наук 13.00.08. Тольятти, 2007. 27 с.

12. Шемет О.В. *Дидактические основы компетентностно ориентированного инженерного образования*: автореферат диссертации на соискание ученой степени доктора педагогических наук 13.00.08. Калуга, 2010. 40 с.
13. Лапо А.И. Практико-ориентированные задачи в курсе информатики. *Педагогическая информатика*. 2018;1:43–54.
14. Окулов С.М. *Информатика: развитие интеллекта школьников*. Москва: БИНОМ. Лаборатория знаний, 2008. 212 с.
15. Окулов С.М. *Развитие интеллекта школьника как принцип организации синергетической среды обучения информатике*: автореферат диссертации на соискание ученой степени доктора педагогических наук 13.00.02. Киров, 2004. 56 с.
16. Лапо А.И. Методика построения систем задач по программированию в школьном курсе информатики. *Весці БДПУ, Сер. 3: Фізіка. Матэматыка. Інфарматыка. Біялогія. Геаграфія*. 2014;4:50–56.
17. Котов В.М., Лапо А.И., Войтехович Е.Н. *Информатика: учеб. пособие для 7 класса учреждений общ. сред. образования с рус. яз. обучения*. Минск: Народная асвета, 2017. 176 с.
18. Макарова Н.П., Лапо А.И., Войтехович Е.Н. *Информатика: учеб. пособие для 6 класса учреждений общ. сред. образования с рус. яз. обучения* Минск: Народная асвета, 2018. 168 с.
19. Котов В.М., Лапо А.И., Быкадоров Ю.А., Войтехович Е.Н. *Информатика: учеб. пособие для 8 класса учреждений общ. сред. образования с рус. яз. обучения*. Минск: Народная асвета, 2018. 168 с.
20. Котов В.М., Лапо А.И., Быкадоров Ю.А., Войтехович Е.Н. *Информатика: учеб. пособие для 9 класса учреждений общ. сред. образования с рус. яз. обучения*. Минск: Народная асвета, 2019. 168 с.
21. Зенько С.И., Быкадоров Ю.А., Казаченок В.В. и др. *Информатика. 6–7 классы*. Дидактические и диагностические материалы: пособие для учителей учреждений общ. среднего образования с бел. и рус. языками обучения (компетентностный подход); под ред. С. И. Зенько. Мозырь: Выснова, 2019. 171 с.
22. Зенько С.И., Быкадоров Ю.А., Казаченок В.В. и др. *Информатика. 8–9 классы*. Дидактические и диагностические материалы: пособие для учителей учреждений общ. среднего образования с бел. и рус. языками обучения (компетентностный подход); под ред. С. И. Зенько. Мозырь: Выснова, 2019. 191 с.

References (на английском языке)

1. Hutorskoj A. Klyucheveye kompetencii kak komponent lichnostno-orientirovannogo obrazovaniya [Key competencies as a component of personality-oriented education]. *Narodnoe obrazovanie*. 2003;2:58–64. (In Russian)
2. Hutorskoj A. Klyucheveye kompetencii. Tekhnologiya konstruirovaniya [Key competencies. Construction Technology]. *Narodnoe obrazovanie*. 2003;5:55–61. (In Russian)
3. Zimnyaya I.A. Klyuchevyye kompetentsii – novaya paradigma rezul'tata sovremennogo obrazovaniya [Key competencies – a new paradigm of the result of modern education]. *Ekspерiment i innovatsiya v shkole*. 2009;2:7–14. (In Russian)
4. Shestakov A.P. Kompetentnostnyy podkhod v obuchenii informatike: kontrol'no-izmeritel'nyye materialy [The competency-based approach to teaching computer science: test and measurement materials]. *Informatika i obrazovaniye*. 2010;6:57–65. (In Russian)
5. Adol'f V.A., Stepanova I. Y. Metodologicheskiye podkhody k formirovaniyu informatsionnoy kul'tury pedagoga [Methodological approaches to the formation of the information culture of the teacher]. *Informatika i obrazovaniye*. 2006;1:2–5. (In Russian)
6. Yeliseyev I.N. Metodologiya otsenki urovnya kompetentsiy studenta [Methodology for assessing the level of student competencies]. *Informatika i obrazovaniye*. 2012;4:80–85. (In Russian)
7. Kondurar M.V. Razvitiye algoritmicheskoy kompetentnosti pri integrirovannom izuchenii diskretnoy matematiki i informatiki studentami kolledzha [The development of algorithmic competence in the integrated study of discrete mathematics and computer science by college students]. *Vektor nauki TGU*. 2014;1:235–238. (In Russian)
8. Lobanova N.V., Man'shin M.E., Smykovskaya T.K. Struktura i formirovaniye intellektual'noy kompetentnosti budushchego uchitelya s dopolnitel'noy spetsial'nost'yu «informatika» [The structure and formation of the intellectual competence of a future teacher with an additional specialty "computer science"]. *Sredneye professional'noye obrazovaniye*. 2010;3:42–46. (In Russian)
9. Man'shin M.E., Babenko A.A. Formirovaniye algoritmicheskoy kompetentnosti budushchego uchitelya informatiki pri izuchenii baz dannykh i sistem upravleniya bazami dannykh [The formation of the algorithmic competence of the future teacher of computer science in the study of databases and database management systems]. *Grani poznaniya: elektronnyy nauchno-obrazovatel'nyy zhurnal VGSPU*. [Elektronnyy resurs]. 2012;6(20). Available at: <http://www.grani.vspu.ru/jurnal/25> (date of access: 09.12.2019). (In Russian)
10. Yarygin O.N. Metodologiya formirovaniya kompetentnosti v analiticheskoy deyatel'nosti pri podgotovke nauchnykh i nauchno-pedagogicheskikh kadrov: avtoreferat

dissertatsii na soiskaniye uchenoy stepeni doktora pedagogicheskikh nauk 13.00.08. [*The methodology of the formation of competence in analytical activities in the preparation of scientific and scientific-pedagogical personnel: abstract of dissertation for the degree of Doctor of Pedagogical Sciences 13.00.08.*]. Tol'yatti, 2012. 42 p. (In Russian)

11. Yarygin O.N. Formirovaniye intellektual'noy kompetentnosti studentov IT-spetsial'nostey v protsesse izucheniya diskretnoy matematiki: avtoreferat dissertatsii na soiskaniye uchenoy stepeni kandidata pedagogicheskikh nauk 13.00.08. [*The formation of intellectual competence of students of IT specialties in the process of studying discrete mathematics: abstract of dissertation for the degree of candidate of pedagogical sciences 13.00.08.*]. Tol'yatti, 2007. 27 p. (In Russian)

12. Shemet O.V. Didakticheskiye osnovy kompetentnostno oriyentirovannogo inzhenernogo obrazovaniya: avtoreferat dissertatsii na soiskaniye uchenoy stepeni doktora pedagogicheskikh nauk 13.00.08. [*Didactic foundations of competency-based engineering education: abstract of dissertation for the degree of Doctor of Pedagogical Sciences 13.00.08.*]. Kaluga, 2010. 40 p. (In Russian)

13. Lapo A.I. Praktiko-oriyentirovannyye zadachi v kurse informatiki [Practice-oriented tasks in the course of computer science]. *Pedagogicheskaya informatika*. 2018;1:43–54. (In Russian)

14. Okulov S.M. Informatika: razvitiye intellekta shkol'nikov [*Informatics: the development of intelligence of schoolchildren*]. Moscow: BINOM. Laboratoriya znaniy, 2008. 212 p. (In Russian)

15. Okulov S.M. Razvitiye intellekta shkol'nika kak printsip organizatsii sinergeticheskoy sredy obucheniya informatike: avtoreferat dissertatsii na soiskaniye uchenoy stepeni doktora pedagogicheskikh nauk 13.00.02. [*The development of schoolchild intelligence as a principle of organizing a synergetic learning environment for computer science: abstract of dissertation for the degree of Doctor of Pedagogical Sciences 13.00.02.*]. Kirov, 2004. 56 p. (In Russian)

16. Lapo A.I. Metodika postroyeniya sistem zadach po programmirovaniyu v shkol'nom kurse informatiki [A technique for constructing systems of programming problems in a school course in computer science]. *Viesci BDPU. Ser. 3: Fizika. Matematyka. Infarmatyka. Biyalogiya. Geografiya*. 2014;4:50–56. (In Russian)

17. Kotov V.M., Lapo A.I., Voytekhovich E.N. Informatika: Ucheb. posobiye dlya 7 klassa uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz [*Informatics: Textbook. allowance for grade 7 institutions total. wednesday education with Russian lang. training*]. Minsk: Narodnaya asveta, 2017. 176 p. (In Russian)

18. Mararova N.P., Lapo A.I., Voytekhovich E.N. Informatika: Ucheb. posobiye dlya 6 klassa uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz [*Informatics: Textbook. allowance for grade 6 institutions total. wednesday education with Russian lang. training*]. Minsk: Narodnaya asveta, 2018. 168 p. (In Russian)
19. Kotov V.M., Lapo A.I., Bykadorov Y.A., Voytekhovich E.N. Informatika: Ucheb. posobiye dlya 8 klassa uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz [*Informatics: Textbook. allowance for grade 8 institutions total. wednesday education with Russian lang. training*]. Minsk: Narodnaya asveta, 2018. 168 p. (In Russian)
20. Kotov V.M., Lapo A.I., Bykadorov Y.A., Voytekhovich E.N. Informatika: Ucheb. posobiye dlya 9 klassa uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz [*Informatics: Textbook. allowance for grade 9 institutions total. wednesday education with Russian lang. training*]. Minsk: Narodnaya asveta, 2019. 168 p. (In Russian)
21. Zen'ko S.I., Bykadorov Y.A., Kazachenok V.V. i dr. Informatika. 6–7 klassy. Didakticheskiye i diagnosticheskiye materialy: posobiye dlya uchiteley uchrezhdeniy obshch. srednego obrazovaniya s bel. i rus. yazykami obucheniya (kompetentnostnyy podkhod); pod red. S. I. Zen'ko. [*Computer science. Grades 6-7. Didactic and diagnostic materials: a manual for teachers of institutions total. secondary education with bel. and Russian languages of instruction*]. Mozyr': Vysnova, 2019. 171 p. (In Russian)
22. Zen'ko S.I., Bykadorov Y.A., Kazachenok V.V. i dr. Informatika. 8–9 klassy. Didakticheskiye i diagnosticheskiye materialy: posobiye dlya uchiteley uchrezhdeniy obshch. srednego obrazovaniya s bel. i rus. yazykami obucheniya (kompetentnostnyy podkhod); pod red. S. I. Zen'ko. [*Computer science. Grades 8-9. Didactic and diagnostic materials: a manual for teachers of institutions total. secondary education with bel. and Russian languages of instruction*]. Mozyr': Vysnova, 2019. 191 p. (In Russian)

Приложение к статье Лапо А.И.
«Компетентностный подход в обучении учащихся программированию».

**Система задач (задачный комплекс)
по теме «Команды ветвления», VIII класс**

Раздел А содержит задачи типов А и В, **раздел В** содержит задачи типов С–Е, предназначенных для индивидуальной работы учащихся, **раздел С** содержит задачи типа Е.

Раздел А

Задание 1. Написать программу, которая выдаст на экран значение *True* или *False*, в зависимости от того, является ли введенное число *x* положительным или нет.

Этапы выполнения задания.

- I Определение исходных данных: переменная *x* (введенное число).
- II Определение результатов: переменная *a* (логическая переменная).
- III Алгоритм решения задачи.
 - 1 Ввод исходных данных.
 - 2 Вычисление значения логической переменной. Число является положительным, если оно больше нуля. Значение переменной *a* будет определяться значением логического выражения $x > 0$.
 - 3 Вывод результата.
- IV Описание переменных:
Pascal: переменная *x* – *integer* (или *real*, для данной задачи это не имеет значения), переменная *a* – *boolean*.
C++: переменная *x* – *int* (или *double*, для данной задачи это не имеет значения), переменная *a* – *bool*.
- V Программа Pascal:

```
Var x: integer;  
    a: boolean;  
Begin  
    Write('введите значение x=');  
    Readln(x);  
    a := x > 0;  
    Write('Число положительное - ', a);  
End.
```
- VI Программа C++:

```
#include <iostream>  
using namespace std;  
int main()
```

```

{
  int x;
  cout << "vvedite znachenie x=";
  cin >> x;
  bool a;
  a = x > 0;
  cout << "Chislo polozhitelnoe - " << a << endl;
  return 0;
}

```

VII Тестирование

1 Запустите программу и введите значения

x=3

Проверьте, результат должен быть следующим:

Число положительное – True

2 Заполните таблицу

	<i>x</i>	<i>a</i>
1	5	
2	0	
3	-13	

3 В каком случае программа выдает значение *False*?

4 Сформулируйте обратную задачу: что бы для всех тех случаев, для которых в исходной задаче получалось *True*, было бы *False* и, наоборот, для всех тех случаев, в которых в исходной задаче получалось *False*, было бы *True*.

5 Внесите изменения в программу, для решения обратной задачи.

Задание 2. Написать программу, которая выдаст на экран значение *True* или *False*, в зависимости от того, является ли введенное число *x* четным или нет.

Этапы выполнения задания.

I Определение исходных данных: переменная ____ (введенное число).

II Определение результатов: переменная ____ (логическая переменная).

III Алгоритм решения задачи.

1 Ввод исходных данных

2 Вычисление значения логической переменной. Число является четным, если остаток от деления данного числа на 2 равен ____ . Значение переменной *t* будет определяться значением логического выражения $___ \bmod 2 = 0$ ($___ \% 2 == 0$).

3 Вывод результата.

IV Описание переменных:

Pascal: Переменная ____ – *integer*, переменная ____ – *boolean*.

C++: Переменная ____ – *int*, переменная ____ – *bool*

V Программа на Pascal:

```

Var ____: integer;
    ____: boolean;
Begin

```

```

Write('введите значение ___=' );
Readln(___);
___ := ___ mod 2 = 0;
Write('Число четное - ', ___);

```

End.

VI Программа на C++:

```

#include <iostream>
using namespace std;
int main()
{
    int ___;
    cout << "vvedite znachenie x=";
    cin >> ___;
    bool ___;
    ___ = x % 2 == 0;
    cout << "Chislo chetnoe - " << ___ << endl;
    return 0;
}

```

VII Тестирование

- 1 Запустите программу и введите значения

$x=2$

Проверьте, результат должен быть следующим:

Число четное - True

- 2 Заполните таблицу

	x	t
1	5	
2	0	
3	-13	
4	-8	

- 3 В каком случае программа выдает значение *False*?
- 4 Сформулируйте обратную задачу: что бы для всех тех случаев, для которых в исходной задаче получалось *True*, было бы *False* и, наоборот, для всех тех случаев, в которых в исходной задаче получалось *False*, было бы *True*.
- 5 Внесите изменения в программу, для решения обратной задачи.
- 6 Запишите решение задачи с использованием команды ветвления. Программа должна выводить на печать сообщения «Число четное» или «Число нечетное».
- 7 * В Паскале есть функция $odd(x)$, которая выдает значение *True*, если число нечетно и *False*, если число четное. Измените программу для решения обратной задачи с использованием функции odd .

Задание 3. Задано число x . Определить, является ли число x положительным или нет.

Выдать на экран соответствующее сообщение. Индивидуальные задания, аналогичные данному размещены в разделе В, задание 1.

Этапы выполнения задания.

- I Определение исходных данных: переменная x (введенное число).
- II Определение результатов: сообщение, зависящее от значения x .
- III Алгоритм решения задачи.
 - 1 Ввод исходных данных
 - 2 Результат будет зависеть от того, истинным или ложным является значение логического выражения $x > 0$. Для решения воспользуемся условным оператором.
 - 3 Вывод результата.
- IV Описание переменных:
 Pascal: переменная x – *integer* (или *real*, для данной задачи это не имеет значения).
 C++: переменная x – *int* (или *double*, для данной задачи это не имеет значения).

V Программа на Pascal:

```

Var x: integer;
Begin
    Write('введите значение x=');
    Readln(x);
    if x>0 then
        Writeln('Число положительное')
    else
        Writeln('Число не положительное');
End.
  
```

VI Программа на C++:

```

#include <iostream>
using namespace std;
int main()
{
    int x;
    cout << "vvedite znachenie x=";
    cin >> x;
    if (x > 0)
        cout << "Chislo pologitelnoe" << endl;
    else
        cout << "Chislo ne pologitelnoe" << endl;
    return 0;
}
  
```

VII Тестирование

- 1 Запустите программу и введите значения

$x=2$

Проверьте, результат должен быть следующим:

Число положительное

- 2 Заполните таблицу

	x	Сообщение
1	18	
2	0	
3	-4	

- 3 Можно ли после слова **else** написать команду `Writeln('Число отрицательное')`? Почему?

- 4 Можно ли заменить логическое выражение в условном операторе так, чтобы сообщения 'Число положительное' и 'Число не положительное' пришлось поменять местами? Как это сделать?
- 5 * Какие изменения нужно внести в программу, чтобы рассматривались три случая: 'Число положительное', 'Число отрицательное', 'Число равно нулю'.

Задание 4. Задано положительное число x . Определить, является это число двузначным или нет. Выдать на экран соответствующее сообщение. Индивидуальные задания, аналогичные данному размещены в разделе В, задание 2.

Этапы выполнения задания.

- I Определение исходных данных: переменная x (введенное число).
- II Определение результатов: сообщение, зависящее от значения x .
- III Алгоритм решения задачи.
 - 1 Ввод исходных данных
 - 2 Число является двузначным, если оно больше 9 и меньше 100. Для решения воспользуемся условным оператором. Условие проверки числа будет составным.
 - 3 Вывод результата.
- IV Описание переменных:

Pascal: переменная x – *integer* (или *real*, для данной задачи это не имеет значения).

C++: переменная x – *int* (или *double*, для данной задачи это не имеет значения).
- V Программа на Pascal:


```

Var x: integer;
Begin
  Write('введите значение x=');
  Readln(x);
  if (x>9) and (x<100) then
    Writeln('Число двузначное')
  else
    Writeln('Число не двузначное');
End.

```
- VI Программа на C++:


```

#include <iostream>
using namespace std;
int main()
{
  int x;
  cout << "vvedite znachenie x=";
  cin >> x;
  if (x > 9 && x < 100)
    cout << "Chislo dvuznachnoe" << endl;
  else
    cout << "Chislo ne dvuznachnoe" << endl;
}

```



```

    return 0;
}

```

VII Тестирование

1 Запустите программу и введите значения

$x=23$

Проверьте, результат должен быть следующим:

Число двузначное

2 Заполните таблицу

	x	Сообщение
1	18	
2	1	
3	124	

3 Измените составное условие в команде ветвления так, чтобы использовались знаки \geq и \leq .

4 Можно ли заменить логическое выражение в условном операторе так, чтобы сообщения 'Число двузначное' и 'Число не двузначное' пришлось поменять местами? Как это сделать?

5 * Какие изменения нужно внести в программу, чтобы рассматривались три случая: 'Число однозначное', 'Число двузначное', 'В числе больше двух цифр'.

Задание 5. Написать программу для решения задачи. Задано число x . Если число четное, то нарисовать на экране зеленый прямоугольник, а если нечетное, то красный круг. Индивидуальные задания, аналогичные данному размещены в разделе В, задание 4.

Этапы выполнения задания.

I Определение исходных данных: переменная x (введенное число).

II Определение результатов: рисунок, зависящий от значения x .

III Алгоритм решения задачи.

1 Ввод исходных данных

2 Результат будет зависеть от того, истинным или ложным является значение логического выражения $x \bmod 2=0$. Для решения воспользуемся условным оператором.

3 Вывод результата.

IV Описание переменных:

Переменная x – *integer*.

V Программа:

```

uses graphABC;
var x:integer;
begin
  write('введите значение x=');
  readln(x);
  writeln(x);

```

```

if x mod 2=0 then
begin
    setBrushColor (clGreen);
    rectangle(150,150,450,350);
end
else
begin
    setBrushColor (clRed);
    circle(400,300,100);
end;
end.

```

VI Тестирование

1 Запустите программу и введите значения

$x=2$

Проверьте, результат должен быть следующим:



2 Заполните таблицу

	1	2	3
x	13	0	-25
рисунок			

- 3 Найдите в программе составные операторы и объясните, зачем они здесь нужны.
- 4 Какие минимальные изменения нужно внести в программу, что бы для четных чисел рисовался красный круг, а для нечетных – зеленый квадрат?
- 5 Внесите в программу изменения, так что бы круг и прямоугольник рисовались случайным цветом.
- 6 *Внесите в программу изменения, так что бы пользователю задавался один из вопросов: 'введите размеры прямоугольника' или 'введите радиус круга'.
- 7 **Какие изменения нужно внести в программу, что бы можно было задавать цвет фигур?

Задание 6. Написать программу для решения задачи (Заполните пропуски). Заданы два различных числа. Меньшее из них заменить их полусуммой, а большее их удвоенным произведением.

Этапы выполнения задания.

- I Определение исходных данных: переменные x и ___ (заданные числа).
- II Определение результатов: числа, удовлетворяющие условию задачи.
- III Алгоритм решения задачи.

- 1 Ввод исходных данных
- 2 Результат будет зависеть от того, истинным или ложным является значение логического выражения _____. Для решения воспользуемся условным оператором.
- 3 Вывод результата.

IV Описание переменных:

Переменные x , _____

V Программа:

```

Var  $x$ , _____ : integer;
Begin
  Write('введите значение  $x$ =');
  Readln( $x$ );
  Write('введите значение ___=');
  Readln(_____);
   $s := (x+y) \text{ div } 2$ ;
   $p := 2 * x * y$ ;
  if _____ then
  begin
     $x := s$ ;
     $y :=$  _____
  end
  else
  begin
    _____ ;
    _____ ;
  end;
  writeln('x=', _____);
End.

```

VI Тестирование

Запустите программу, введите значения и проверьте результаты

Ввод x	Ввод y	Вывод x	Вывод y
2	4	3	6
1	9		
12	4		
13	7		

Задание 7. Вычислить значение функции для заданного x . Индивидуальные задания, аналогичные данному размещены в разделе В, задание 5.

Образец выполнения задания.

$$y = \begin{cases} -2, & \text{если } x < -2 \\ x^3, & \text{если } -2 \leq x \leq 2 \\ x + 2, & \text{если } x > 2 \end{cases}$$

Этапы выполнения задания.

- I Определение исходных данных: переменная x (значение переменной).
- II Определение результатов: переменная y (значение функции).
- III Алгоритм решения задачи.
 - 1 Ввод исходных данных

2 Проверка значения x и вычисление значения функции, в зависимости от значения аргумента

3 Вывод результата.

IV Описание переменных:

Все переменные, определенные для решения задачи, имеют тип *real* (для программы на C++ - тип *double*).

V Программа на Pascal:

```
Var x, y: real;
Begin
  Write('введите значение x=');
  Readln(x);
  If x<-2 then
    y:=-2
  else
    if x<=2 then
      y:=x*x*x
    else
      y:=x+2;
  Writeln('y=',y);
End.
```

VI Программа на C++:

```
#include <iostream>
using namespace std;
int main()
{
  double x, f;
  cout << "vvedite znachenie x=";
  cin >> x;
  if (x < -2)
    f = -2;
  else
    if (x <= 2)
      f = x * x * x;
    else
      f = x + 2;
  cout << "f=" << f << endl;
  return 0;
}
```

VII Тестирование

1 Запустите программу, введите значения и проверьте результаты

x	y
-5	-2
0.2	0.008
3.27	5.27

Проверить правильность вычислений можно на калькуляторе.

2 Заполните таблицу

	x	y
1	-3.52	
2	-1.28	
3	23.48	

- 3 Добавьте в таблицу свои значения x . Предусмотрите все возможные случаи вычисления значения функции. Сколько строк нужно добавить в таблицу?
- 4 Для заданной функции и фрагмента программы на Pascal, вычисляющей значение этой функции, вставьте пропущенное

$$y = \begin{cases} x, & \text{если } x < -3 \\ 2x + \dots, & \text{если } \dots \leq x \leq 7 \\ \dots, & \text{если } x > \dots \end{cases}$$

```

if x<... then
  y:=x
else
  if x<=7 then
    ...:=2*x+7
  else
    y:=2*x-5;

```

- 5 Для заданной функции и фрагмента программы на C++, вычисляющей значение этой функции, вставьте пропущенное

$$y = \begin{cases} x, & \text{если } x < -3 \\ 2x + \dots, & \text{если } \dots \leq x \leq 7 \\ \dots, & \text{если } x > \dots \end{cases}$$

```

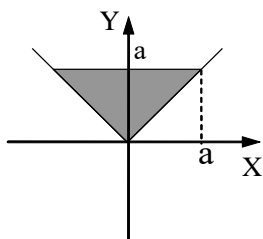
if (x<...)
  y=x;
else
  if (x<=7)
    ...=2*x+7;
  else
    y=2*x-5;

```

- 6 Запишите функцию, значение которой вычисляется с помощью предложенного фрагмента программы.

Pascal	C++
<pre> if x < -3 then y := -x else if x < 0 then y := 0 else y := x * x; </pre>	<pre> if (x < -3) y = -x; else if (x < 0) y = 0; else y = x * x; </pre>

Задание 8. Определить принадлежит ли точка $A(x, y)$ данной области. Ответом должно служить слово "да" или "нет". Индивидуальные задания, аналогичные данному размещены в разделе В, задание 6.



Этапы выполнения задания.

- I Определение исходных данных: переменные x и y (координаты точки).
- II Определение результатов: слово "да" или "нет", в зависимости от принадлежности точки закрашенной области.
- III Алгоритм решения задачи.
 - 1 Ввод исходных данных
 - 2 Область образована тремя прямыми: $y = x$; $y = -x$; $y = a$. Точка принадлежит области, если одновременно выполняются три условия: $y \geq x$,

$y \geq -x$, $y \leq a$. Для того, что бы определить, какой из знаков поставить, достаточно взять произвольную точку, принадлежащую области и посмотреть, как будут выполняться условия для этой точки. Например, при $a = 5$, точка с координатами (1; 3) принадлежит области, и выполняются следующие условия $3 \geq 1$, $3 \geq -1$, $3 < 5$.

3 Вывод результата.

IV Описание переменных: все переменные, определенные для решения задачи, имеют тип *real* (для программы на C++ - тип *double*).

V Программа на Pascal:

```

Var x, y, a: real;
Begin
  Write('введите значения координаты x=');
  Readln(x);
  Write('введите значения координаты y=');
  Readln(y);
  Write('введите значения a=');
  Readln(a);
  If (y>=x) and (y>=-x) and (y<=a) then
    Writeln('да')
  else
    Writeln('нет');
End.

```

VI Программа на C++:

```

#include <iostream>
using namespace std;
int main()
{
  double x, y, a;
  cout << "vvedite x=";
  cin >> x;
  cout << "vvedite y=";
  cin >> y;
  cout << "vvedite a=";
  cin>>a;
  if (y >= x && y >= -x && y <= a)
    cout << "Da" << endl;
  else
    cout << "Net" << endl;
  return 0;
}

```

VII Тестирование

1 Запустите программу, введите значения и проверьте результаты

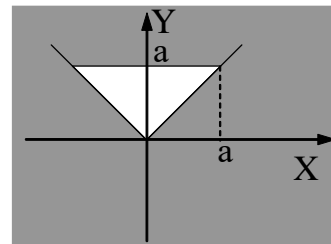
x	y	a	результат
1	3	5	да
-3	1	5	нет
-2	-2	5	нет

2 Заполните таблицу

	x	y	a	результат
1	-1.3	2.8	4	
2	0.5	10.2	8	

3	0	0	5	
---	---	---	---	--

- 3 Добавьте в таблицу свои значения. Предусмотрите различные возможные случаи расположения точки относительно области.
- 4 Проверьте точки, лежащие на границе области. Какой ответ выдает программа? Какие изменения нужно внести в программу, если считать, что точки, лежащие на границе, не принадлежат области?
- 5 Как изменится условие принадлежности точки области, если областью считать все плоскость, за исключением рассмотренного «треугольника».



Задание 9. Написать программу для решения задачи. Проверить является ли введенное число трехзначным, и если да, то вывести четные цифры числа. Индивидуальные задания, аналогичные данному размещены в разделе В, задание 7.

Этапы выполнения задания.

- I Определение исходных данных: переменная a (трехзначное число).
- II Определение результатов: переменные $a1$, $a2$, $a3$, в том случае если они четные или сообщение, что четных цифр нет.
- III Алгоритм решения задачи.
 - 1 Ввод исходного числа.
 - 2 Проверка является ли число трехзначным. Трехзначное число больше 99 и меньше 1000.
 - 3 Если число трехзначное, то выделяем цифры числа и проверяем каждую из них на четность.
 - а) Для выделения первой цифры (переменная $a1$) трехзначного числа необходимо найти целую часть от деления числа на 100.
 - б) Для выделения второй цифры (переменная $a2$) трехзначного числа необходимо найти остаток от деления числа на 100 и от него найти целую часть при делении на 10.
 - в) Для определения последней цифры (переменная $a3$) трехзначного числа необходимо найти остаток от деления числа на 10.
 - г) Для проверки цифры на четность нужно проверить, равен ли нулю остаток от деления цифры на 2.
 - 4 Вывод результата.
- IV Описание переменных:

Все переменные, определенные для решения задачи, имеют тип *integer* (для программы на C++ - тип *int*).
- V Программа на Pascal:


```

Var a, a1, a2, a3: integer;
Begin
      
```

```

Write('введите значение a=');
Readln(a);
If (a>99) and (a<1000) then
Begin
  a1:=a div 100; //первая цифра
  a2:=a mod 100 div 10; //вторая цифра
  a3:=a mod 10; //третья цифра
  if a1 mod 2=0 then
    Writeln('цифра ',a1, ' четная');
  if a2 mod 2=0 then
    Writeln('цифра ',a2, ' четная');
  if a3 mod 2=0 then
    Writeln('цифра ',a3, ' четная');
  If (a1 mod 2=1) and (a2 mod 2=1) and (a3 mod 2=1) then
    Writeln('в числе нет четных цифр');
End
Else
  Writeln('число не является трехзначным');
End.

```

VI Программа на C++:

```

#include <iostream>
using namespace std;
int main()
{
  int a, a1, a2, a3;
  cout << "vvedite a=";
  cin >> a;
  if (a > 99 && a < 1000)
  {
    a1 = a / 100;
    a2 = a % 100 / 10;
    a3 = a % 10;
    if ( a1 % 2 == 0)
      cout << "cifra " << a1 << " chetnaja" << endl;
    if (a2 % 2 == 0)
      cout << "cifra " << a2 << " chetnaja" << endl;
    if (a3 % 2 == 0)
      cout << "cifra " << a3 << " chetnaja" << endl;
    if (a1 % 2 == 1 && a2 % 2 == 1 && a3 % 2 == 1)
      cout<<"v chisle net chetnyh cifr"<<endl;
  }
  else
    cout<<"chislo ne trehznachoe"<<endl;
  return 0;
}

```

VII Тестирование

1 Проверьте работу программы для следующих значений

Ввод <i>a</i>	Вывод
246	цифра 2 четная цифра 4 четная цифра 6 четная
103	цифра 0 четная
537	в числе нет четных цифр
25	число не является трехзначным

2 Заполните таблицу

	<i>a</i>	Вывод
1	468	
2	476	
3	233	
4	441	
5	119	
6	1111	

3 Какие случаи расположения цифр в трехзначном числе не были рассмотрены? Предложите свои тесты на эти случаи.

4 Как по-другому можно вычислить значение второй цифры трехзначного числа?

5 Можно ли проверку цифр в числе записать следующим образом:

```

if a1 mod 2=0 then
    Writeln('цифра ',a1, ' четная')
else
    if a2 mod 2=0 then
        Writeln('цифра ',a2, ' четная')
    else
        if a3 mod 2=0 then
            Writeln('цифра ',a3, ' четная')
        else
            Writeln('в числе нет четных цифр');

```

Аргументируйте свой ответ.

Задание 10. Написать программу для решения задачи с использованием оператора *case* (*switch*). Вводится время в часах от 0 до 23. Определить к какой части суток относится данное время. Индивидуальные задания, аналогичные данному размещены в разделе В, задание 10.

Этапы выполнения задания.

I Определение исходных данных: переменная *t* (время).

II Определение результатов: сообщение утро, день, вечер или ночь в зависимости от введенного времени.

III Алгоритм решения задачи.

1. Ввод исходного числа.

2. Выбор значения переменной *t* из возможных диапазонов времени:

а) от 23 до 5 – ночь

б) от 6 до 10 – утро

в) от 11 до 17 день

г) от 18 до 22 вечер.

3. Вывод результата.

IV Описание переменных:

Переменная *t* имеет тип *integer*.

V Программа на Pascal:

```

Var t: integer;
Begin
  Write('введите значение t=');
  Readln(t);
  Case t of
    23, 0..5: Writeln('Ночь');
    6..10: Writeln('Утро');
    11..17: Writeln('День');
    18..22: Writeln('Вечер')
    else Writeln('ошибочный ввод');
  end;
End.

```

VI Программа на C++:

```

#include <iostream>
using namespace std;
int main()
{
  int t;
  cout<<"vvedite znachenie t=";
  cin>>t;
  switch (t)
  {
    case 23: case 0: case 1: case 2: case 3: case 4: case 5:
      cout << "Noch" << endl; break;
    case 6: case 7: case 8: case 9: case 10:
      cout << "Utro" << endl; break;
    case 11: case 12: case 13: case 14: case 15: case 16: case 17:
      cout << "Den" << endl; break;
    case 18: case 19: case 20: case 21: case 22:
      cout << "Vecher" << endl; break;
    default:
      cout << "Oshibocny vvod" << endl;
  }
  return 0;
}

```

VII Тестирование

- 1 Проверьте работу программы для следующих значений

Ввод t	Вывод
2	Ночь
18	Вечер

- 2 Заполните таблицу, предусмотрев проверку всех возможных ситуаций. Сколько строк должно быть в таблице?

	t	Вывод

- 3 Как будет реагировать программа на значение $t=30$, если из программы убрать блок *else*?
- 4 В каких случаях следующая программа решения этой же задачи будет работать некорректно?

```

Var t: integer;
Begin

```

```

Write('введите значение t=');
Readln(t);
If (t<0) or (t>23) then
    Writeln('ошибочный ввод')
else
    Case t of
        23, 0..5: Writeln('Ночь');
        6..10: Writeln('Утро');
        11..17: Writeln('День');
        else Writeln('Ночь');
    end;
End.

```

Аргументируйте свой ответ.

5 Решите данную задачу с использованием неполной записи оператора *case*.

Раздел В

Задание 1. Написать программу для решения задачи. Разбор типового варианта можно посмотреть в разделе А, задание 3.

1. Задано положительное число x , являющееся возрастом человека в годах. Определить, человек совершеннолетний или нет. Выдать на экран соответствующее сообщение.
2. Задано неотрицательное число x . Определить, может ли это число быть школьной отметкой или нет. Выдать на экран соответствующее сообщение.
3. Заданы два числа x и y . Определить, верно ли, что первое число больше второго. Выдать на экран соответствующее сообщение.
4. Заданы два числа x и y . Определить, верно ли, что второе число меньше первого. Выдать на экран соответствующее сообщение.
5. Задано число x . Определить, является ли число x кратным 3 или нет. Выдать на экран соответствующее сообщение.
6. Задано число x . Определить, делится ли число без остатка на 10 или нет. Выдать на экран соответствующее сообщение.
7. Задано число x . Определить, делится ли число без остатка на 3 и на 5 или нет. Выдать на экран соответствующее сообщение.
8. Задано число x . Определить, делится ли число без остатка на 2, на 3 и на 5 или нет. Выдать на экран соответствующее сообщение.
9. Заданы два положительных числа x и y . Определить, верно ли, что x делится на y , без остатка или нет. Выдать на экран соответствующее сообщение.
10. Заданы два числа x и y , отличных от нуля. Определить, верно ли, что оба эти числа имеют один знак (или оба положительные или оба отрицательные). Выдать на экран соответствующее сообщение.
11. Заданы два числа x и y , отличных от нуля. Определить, верно ли, что эти числа имеют разные знаки (одно положительное, а другое отрицательное). Выдать на экран соответствующее сообщение.

12. Задано положительное число x . Определить, является это число полным квадратом или нет. Выдать на экран соответствующее сообщение

Задание 2. Написать программу для решения задачи. Разбор типового варианта можно посмотреть в разделе А, задание 4.

1. Задано положительное число x . Определить, является это число трехзначным или нет. Выдать на экран соответствующее сообщение.
2. Задано положительное число x . Определить, является это число четырехзначным или нет. Выдать на экран соответствующее сообщение.
3. Задано положительное число x . Определить, является это число пятизначным или нет. Выдать на экран соответствующее сообщение.
4. Задано положительное число x . Определить, является это число шестизначным или нет. Выдать на экран соответствующее сообщение.
5. Задано положительное число x . Определить, является это число четным двузначным числом или нет. Выдать на экран соответствующее сообщение.
6. Задано положительное число x . Определить, является это число нечетным трехзначным числом или нет. Выдать на экран соответствующее сообщение.
7. Задано положительное число x . Определить, является это число четырехзначным числом кратным пяти или нет. Выдать на экран соответствующее сообщение.
8. Задано положительное число x . Определить, является это число пятизначным числом кратным трем или нет. Выдать на экран соответствующее сообщение.
9. Заданы два положительных числа x и y . Определить, верно ли, что одно из них делится на другое без остатка или нет. Выдать на экран соответствующее сообщение.
10. Заданы два положительных числа x и y . Определить, верно ли, что они оба являются двузначными числами. Выдать на экран соответствующее сообщение.
11. Заданы два положительных числа x и y . Определить, верно ли, что хотя бы одно из этих чисел является трехзначным и их сумма является четной. Выдать на экран соответствующее сообщение.
12. Заданы два положительных числа x и y . Определить, верно ли, что первое число больше второго, и хотя бы одно из них четное. Выдать на экран соответствующее сообщение.

Задание 3. Написать программу для решения задачи.

1. Сгенерировать случайное число x из промежутка $[0; 10)$. Определить, является ли число x числом большим 5 или нет. Выдать на экран соответствующее сообщение.
2. Сгенерировать случайное число x из промежутка $[0; 20)$. Определить, является ли число x числом меньшим 10 или нет. Выдать на экран соответствующее сообщение.
3. Сгенерировать случайное число x из промежутка $[0; 10)$. Определить, является ли число x четным или нет. Выдать на экран соответствующее сообщение.

4. Сгенерировать случайное число x из промежутка $[0; 20)$. Определить, является ли число x кратным 3 или нет. Выдать на экран соответствующее сообщение.
5. Сгенерировать случайное число x из промежутка $[-10; 10)$. Определить, является ли число x положительным или нет. Выдать на экран соответствующее сообщение.
6. Сгенерировать случайное число x из промежутка $[-20; 20)$. Определить, является ли число x отрицательным или нет. Выдать на экран соответствующее сообщение.
7. Сгенерировать случайное число x из промежутка $[-100; 100)$. Определить, является ли число x отрицательным четным числом или нет. Выдать на экран соответствующее сообщение.
8. Сгенерировать случайное число x из промежутка $[-100; 100)$. Определить, является ли число x положительным нечетным числом или нет. Выдать на экран соответствующее сообщение.
9. Дано целое положительное число a . Сгенерировать случайное число x из промежутка $[-a; a)$. Определить, является ли число x положительным числом кратным 7 или нет. Выдать на экран соответствующее сообщение.
10. Дано целое положительное число a . Сгенерировать случайное число x из промежутка $[-a; a)$. Определить, является ли число x отрицательным числом кратным 5 или нет. Выдать на экран соответствующее сообщение.
11. Даны целые числа a и b ($a < b$). Сгенерировать случайное число x из промежутка $[a; b)$. Определить, верно ли что число x меньше середины промежутка $[a; b)$ и кратно 3. Выдать на экран соответствующее сообщение.
12. Даны целые числа a и b ($a < b$). Сгенерировать случайное число x из промежутка $[a; b)$. Определить, верно ли что число x отличается от середины промежутка $[a; b)$ не более чем на $\frac{1}{3}$ длины промежутка. Выдать на экран соответствующее сообщение.

Задание 4. Написать программу для решения задачи. Разбор типового варианта можно посмотреть в разделе А, задание 5.

1. Задано число x . Если число нечетное, то нарисовать на экране овал, в котором горизонтальный радиус больше вертикального, а если четное, овал, в котором вертикальный радиус больше горизонтального.
2. Задано число x . Если число четное, то нарисовать на экране треугольник, а если нечетное, то пятиугольник.
3. Задано число x . Если число четное, то нарисовать на экране горизонтальный отрезок, а если нечетное, то вертикальный отрезок.
4. Задано число x . Нарисовать на экране светофор для пешеходов. Если число x четное, то должен гореть зеленый свет, а если нечетное, то – красный.

5. Задано число x . Нарисовать на экране верхнюю половинку круга, если число x положительное или нижнюю половинку круга, если число x неотрицательное.
6. Задано число x . Нарисовать на экране круг, вписанный в квадрат, если число x положительное или квадрат, вписанный в круг, если число x неотрицательное.
7. Заданы два числа x, y . Если $x > y$, то нарисовать на экране два эллипса, горизонтальный радиус которых больше вертикального, в противном случае нарисовать два эллипса, у которых вертикальный радиус больше горизонтального.
8. Заданы два числа x, y . Если $x > y$, то нарисовать на экране прямоугольник, длина (горизонтальная сторона) которого больше ширины (вертикальной стороны), в противном случае нарисовать прямоугольник, у которого ширина больше длины.
9. Задано число x . Нарисовать на экране компьютера круг с радиусом, равным x . Если $x < 1$ или $x > 200$, то выдать сообщение о том, что круг нарисовать нельзя.
10. Задано число x . Нарисовать на экране компьютера отрезок длиной, равной x . Если $x < 1$ или $x > 600$, то выдать сообщение о том, что отрезок нарисовать нельзя.
11. Заданы два числа x, y . Нарисовать на экране компьютера прямоугольник с длинами сторон, равными x и y . Если $x < 1$, или $x > 600$, или $y < 1$, или $y > 450$, то выдать сообщение о том, что прямоугольник нарисовать нельзя.
12. Заданы два числа x, y . Нарисовать на экране компьютера эллипс с горизонтальным и вертикальным радиусами, равными x и y . Если $x < 1$, или $x > 300$, или $y < 1$, или $y > 220$, то выдать сообщение о том, что эллипс нарисовать нельзя.

Задание 5. Написать программу для решения задачи. В каждом варианте вычислить значение функции для заданного x . Разбор типового варианта можно посмотреть в разделе А, задание 7.

1.
$$y = \begin{cases} x^2, & \text{если } x \geq 2 \\ 4, & \text{если } x < 2 \end{cases}$$
2.
$$y = \begin{cases} x^2 + 4x + 5, & \text{если } x \leq 3 \\ \frac{1}{x^2 + 4x + 5}, & \text{если } x > 3 \end{cases}$$
3.
$$y = \begin{cases} x^2, & \text{если } -1 \leq x \leq 1 \\ |x|, & \text{в остальных случаях} \end{cases}$$
4.
$$y = \begin{cases} x - 7, & \text{если } x < -7 \\ \sqrt{49 - x^2}, & \text{если } -7 \leq x \leq 7 \\ -x + 7, & \text{если } x > 7 \end{cases}$$
5.
$$y = \begin{cases} 0, & \text{если } x < 0 \\ x^2 - x, & \text{если } 0 \leq x \leq 1 \\ x^2 + 2, & \text{если } x > 1 \end{cases}$$
6.
$$y = \begin{cases} \sqrt{x^2 + 4}, & \text{если } x < -2 \\ x^3, & \text{если } -2 \leq x \leq 2 \\ x - 2, & \text{если } x > 2 \end{cases}$$

$$7. \quad y = \begin{cases} 2x^2 - 3, & \text{если } x < 1 \\ 36 - x^2, & \text{если } 1 \leq x \leq 17 \\ 1 - 10x, & \text{если } 1 \leq x \leq 17 \\ -x^3 - 2, & \text{если } x > 17 \end{cases}$$

$$8. \quad y = \begin{cases} |x - 4|, & \text{если } x < -4 \\ x^3 - 2, & \text{если } -4 \leq x \leq 4 \\ x^2 + 4, & \text{если } -4 \leq x \leq 4 \\ 3x - 5, & \text{если } x > 4 \end{cases}$$

$$9. \quad y = \begin{cases} x + 4, & \text{если } x \leq 0 \\ x + 1, & \text{если } 0 < x < 1 \\ 2, & \text{если } 1 \leq x \leq 2 \\ 3x - 4, & \text{если } x > 2 \end{cases}$$

$$10. \quad y = \begin{cases} |x - 1|, & \text{если } x \leq -10 \\ x^2 + 1, & \text{если } -10 < x \leq 0 \\ \sqrt{x \cdot [x]}, & \text{если } 0 < x < 10 \\ \frac{1}{x^2}, & \text{если } x \geq 10 \end{cases}$$

Примечание: запись $[x]$ обозначает целую часть числа x , в Pascal – функция trunc(x)

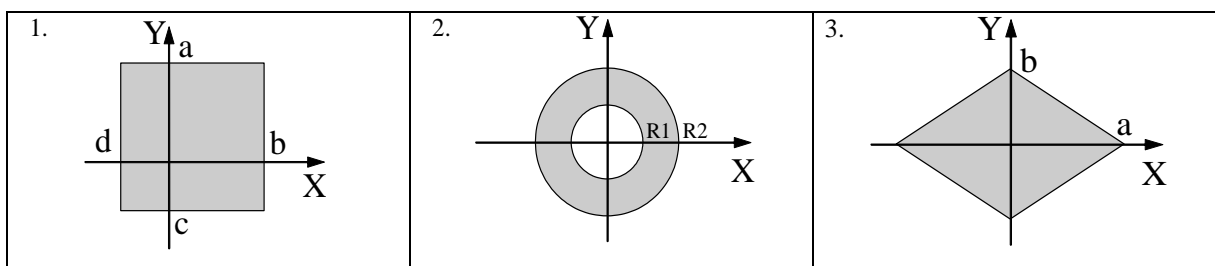
$$11. \quad y = \begin{cases} |x|, & \text{если } x < -1 \\ x^3, & \text{если } -1 \leq x \leq 1 \\ \sqrt{x + [x]}, & \text{если } 1 < x < 10 \\ \frac{x}{x^2 + 4}, & \text{если } x \geq 10 \end{cases}$$

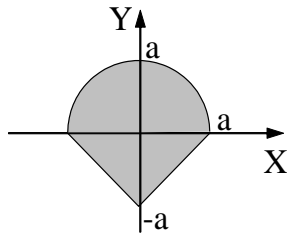
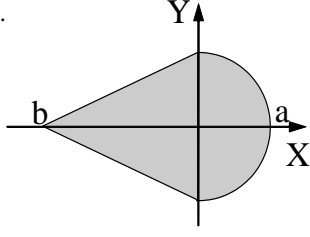
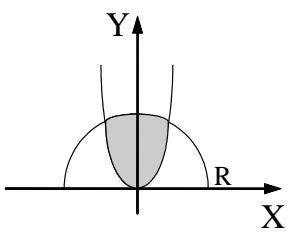
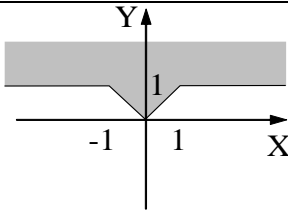
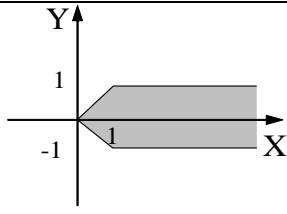
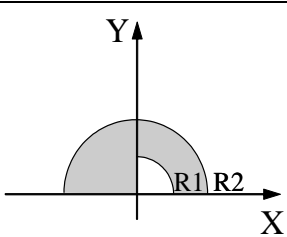
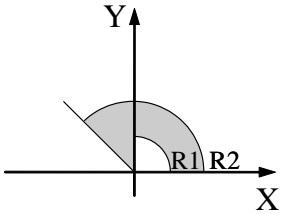
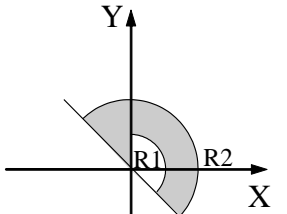
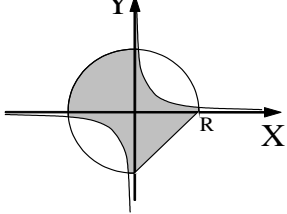
Примечание: запись $[x]$ обозначает целую часть числа x , в Pascal – функция trunc(x)

$$12. \quad y = \begin{cases} -7x^2 + x^3, & \text{если } x < -3 \\ \{x\}, & \text{если } -3 \leq x \leq 3 \\ [x], & \text{если } 3 < x < 33 \\ \frac{x + 1}{x^3}, & \text{если } x \geq 33 \end{cases}$$

Примечание: запись $[x]$ обозначает целую часть числа x , в Pascal – функция trunc(x). Запись $\{x\}$ обозначает дробную часть числа x , в Pascal – функция frac(x)

Задание 6. Написать программу для решения задачи. Для каждого варианта определить принадлежит ли точка $A(x, y)$ данной области. Ответом должно служить слово "да" или "нет". Для каждой области записаны уравнения линий, ограничивающие область. Разбор типового варианта можно посмотреть в разделе А, задание 8.



$y = a ; y = c ;$ $x = b ; x = d$	$x^2 + y^2 = R1^2 ;$ $x^2 + y^2 = R2^2$	$y = -\frac{b}{a}x + b ; y = \frac{b}{a}x + b ;$ $y = -\frac{b}{a}x - b ; y = \frac{b}{a}x - b$
4.  $x^2 + y^2 = a^2$ $y = x - a ;$	5.  $x^2 + y^2 = a^2$ $y = -\frac{a}{b}x + a ; y = \frac{a}{b}x - a$	6.  $x^2 + y^2 = R^2$ $y = x^2$
7.  $y = 1 ; y = x $	8.  $y = 1 ; y = -1 ; y = x$	9.  $x^2 + y^2 = R1^2 ;$ $x^2 + y^2 = R2^2$
10.  $x^2 + y^2 = R1^2 ;$ $x^2 + y^2 = R2^2$ $y = -x$	11.  $x^2 + y^2 = R1^2 ;$ $x^2 + y^2 = R2^2$ $y = -x$	12.  $x^2 + y^2 = R^2 ;$ $y = x - R ; yx = 1$

Задание 6. Написать программу для решения задачи. Разбор типового варианта можно посмотреть в разделе А, задание 9.

1. Дано натуральное число. Проверить, является ли число двузначным, и равна ли первая цифра числа второй.
2. Дано натуральное число. Проверить, является ли число двузначным, у которого первая цифра кратна второй.
3. Дано натуральное число. Проверить, является ли число трехзначным, и равна ли первая цифра числа третьей.
4. Дано натуральное число. Проверить, является ли число трехзначным, и являются ли все цифры числа нечетными.

5. Дано натуральное число. Проверить, является ли оно четырехзначным и содержит ли данное число одинаковые цифры.
6. Дано натуральное число. Проверить, является ли оно четырехзначным палиндромом (т. е. равна ли первая цифра четвертой, а вторая третьей).
7. Дано натуральное число. Проверить, является ли число четырехзначным “счастливым” (т. е. сумма первых двух цифр равна сумме последних двух).
8. Дано натуральное число. Проверить, является ли оно четырехзначным, содержащим ровно три одинаковые цифры.
9. Дано натуральное число. Проверить, является ли оно пятизначным, у которого цифры расположены в порядке возрастания.
10. Дано натуральное число. Проверить, является ли оно пятизначным, у которого первая цифра равна последней, а сумма второй и четвертой является четным числом.
11. Дано натуральное число. Проверить, является ли число шестизначным “счастливым” (т. е. сумма первых трех цифр равна сумме последних трех).
12. Дано натуральное число. Проверить, является ли число шестизначным и выполняется ли следующее условие: сумма первой и четвертой цифр равна сумме второй и пятой и равна сумме третьей и шестой.

Задание 8. Написать программу для решения задачи.

1. Определить, является ли три данных целых числа четными?
2. Определить, имеют ли оба заданных числа одинаковую четность?
3. Посчитать количество четных среди чисел a, b, c .
4. Посчитать количество четных и количество нечетных среди чисел a, b, c .
5. Посчитать количество отрицательных и количество неотрицательных среди трех чисел a, b, c .
6. Определить, чисел какого знака (положительных или отрицательных) больше среди чисел a, b, c .
7. Посчитать количество положительных и количество отрицательных среди четырех чисел a, b, c, d .
8. Определить, имеется ли среди чисел a, b, c хотя бы одна пара равных между собой чисел. Указать какие.
9. Определить, имеется ли среди чисел a, b, c хотя бы одна пара кратных трем чисел. Указать какие.
10. Посчитать среди четырех чисел a, b, c, d , количество чисел, дающих нечетный остаток при делении на 4.
11. Посчитать среди четырех чисел a, b, c, d , количество чисел, дающих четный остаток при делении на 3.
12. Посчитать среди четырех чисел a, b, c, d , количество чисел, дающих нечетный остаток при делении на 5.

Задание 9. Написать программу для решения геометрической задачи.

1. Определить, является ли треугольник со сторонами a, b, c равносторонним. Если "да", то вычислить его площадь.
2. Определить, является ли треугольник со сторонами a, b, c равнобедренным. Если "да", то, что является основанием, а что боковыми сторонами.
3. Определить, является ли треугольник со сторонами a, b, c прямоугольным. Если "да", то, что является гипотенузой, а что катетами.
4. Числа a_1, a_2, a_3 – стороны одного треугольника, а b_1, b_2, b_3 – другого. Определить являются ли треугольники подобными, если – "да", то определить коэффициент подобия.
5. Определить, является ли треугольник со сторонами a, b, c тупоугольным. Если "да", то, какие стороны образуют тупой угол?
6. Даны три числа a, b, c . Определить существует ли треугольник со сторонами a, b, c . Если он существует определить его вид (прямоугольный, остроугольный, тупоугольный).
7. Даны действительные положительные числа a, b, c, d . Выяснить, можно ли построить четырехугольник с такими длинами сторон.
8. Даны действительные положительные числа a, b, c, d . Выяснить можно ли прямоугольник со сторонами a, b уместить внутри прямоугольника со сторонами c, d так, чтобы каждая из сторон одного прямоугольника была параллельна или перпендикулярна каждой стороне второго прямоугольника.
9. Действительные числа $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ являются координатами вершин четырехугольника. Проверить, является ли этот четырехугольник прямоугольником со сторонами, параллельными осям координат.
10. Действительные числа $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$ являются координатами вершин прямоугольника со сторонами, параллельными осям координат. Принадлежит ли начало координат этому прямоугольнику?
11. Дан квадрат с вершинами в точках $(0, 0), (0, 1), (1, 1), (1, 0)$. Вычислить наибольшее из расстояний от точки с координатами (x, y) до вершин данного квадрата.
12. Дан квадрат с вершинами в точках $(0, 0), (0, 1), (1, 1), (1, 0)$. Вычислить наименьшее из расстояний от точки с координатами (x, y) , лежащей внутри квадрата до сторон данного квадрата.

Задание 10. Написать программу для решения задачи с использованием команды case.

Разбор типового варианта можно посмотреть в разделе А, задание 10.

1. Вводится число от 1 до 4, определяющее пору года. Дать название этой поры года. (1 – зима, 2 – весна, 3 – лето, 4 – осень).
2. Вводится число от 1 до 7, определяющее день недели. Дать название этого дня. (1 – понедельник, 2 – вторник, ..., 7 – воскресенье).
3. Вводится число от 1 до 12, определяющее месяц года. Дать название этого месяца года. (1 – январь, 2 – февраль, ..., 12 – декабрь).

4. Вводится число от 1 до 10. Дать название этого числа. (1 – один, 2 – два, ..., 10 – десять).
5. Вводится число от 1 до 12, определяющее месяц года. Определить к какому кварталу принадлежит месяц. (Каждые три месяца года – новый квартал).
6. Вводится число от 1 до 12, определяющее месяц года. Определить к какой поре года принадлежит месяц.
7. Дано натуральное число N ($N < 20$), определяющее сумму денег в рублях. Дать для этого числа наименование: "рубль", "рубля", "рублей".
8. Дано натуральное число N ($N < 100$), определяющее возраст человека в годах. Дать для этого числа наименование: "год", "года", "лет".
9. Вводится число от 1 до 12, определяющее номер класса, в котором обучается ученик или ученица, также вводится пол (0 – женский, 1 – мужской). Дать название ученику(це) в зависимости от класса. Например, если ввели 5 и 0, то это – пятиклассница, если 12 и 1 – двенадцатиклассник.
10. Вводится число от 1 до 20. Дать название этого числа. (1 – один, 2 – два, ..., 20 – двадцать).
11. Вводится число от 1 до 20. Дать название этого числа на иностранном языке.
12. Вводится число от 1 до 100. Дать название этого числа. (1 – один, 2 – два, ..., 100 – сто).

Примечание. Для задач вариантов 10, 11 и 12 необходимо вспомнить, как образуются названия числительных в русском или иностранном языках. (тринадцать, тридцать и т. д.)

Раздел С

Для решения предложенных задач разработать алгоритмы и написать программы.

1. Дан номер года. Определить является ли он високосным (год является високосным, если его номер делится на 4, за исключением тех, которые делятся на 100 и не делятся на 400).
2. Дана тройка чисел a , b , c . Проверить могут ли они быть датой (число, месяц, год). Например, 3, 7, 1972 – дата. Для определения високосного года см. задачу 1.
3. Определить взаимное расположение двух прямых $y = k_1x + b_1$ и $y = k_2x + b_2$ на плоскости. (Совпадают, параллельны или пересекаются). Если прямые пересекаются, то найти координаты точки пересечения.
4. Даны действительные числа x_1 , y_1 , x_2 , y_2 , x_3 , y_3 , являющиеся координатами вершин треугольника. Принадлежит ли начало координат этому треугольнику?
5. Даны три числа. Возвести в квадрат те из них, которые положительны. Остальные оставить без изменения.

6. Даны три числа. Удвоить те из них, которые нечетны. Остальные оставить без изменения.
7. Даны три числа. Удвоить их, если они упорядочены по возрастанию, иначе отрицательные заменить их модулями.
8. Даны три числа. Удвоить их, если они упорядочены по возрастанию, утроить, если – по убыванию, а иначе четные возвести в квадрат.
9. Даны четыре числа. Если первое число четное, то выведите сумму этих чисел. Если сумма второго и третьего числа четная, то выведите среднее арифметическое четырех чисел. Если ни одно из вышеперечисленных условий не выполняется, выведите большее из третьего и четвертого чисел.
10. Даны четыре числа. Если второе и четвертое числа отрицательны, то выведите их произведение. Если первое и третье числа больше второго, то выведите их сумму. Если не выполняется ни одно из выше перечисленных условий, то выведите все четыре числа, возведенные в квадрат.
11. Дано положительное целое трехзначное число. Определить, равен ли квадрат этого числа сумме кубов его цифр. Выдать на экран соответствующее сообщение.
12. Дано положительное целое четырехзначное число. Определить, равно ли произведение цифр числа сумме квадратов его цифр. Выдать на экран соответствующее сообщение.
13. Даны четыре числа – координаты двух полей шахматной доски. Определить может ли ладья переместиться с первого поля на второе. Ладья может перемещаться только по горизонтали или по вертикали.
14. Даны четыре числа – координаты двух полей шахматной доски. Определить может ли слон переместиться с первого поля на второе. Слон может перемещаться только по диагонали.
15. Даны четыре числа – координаты двух полей шахматной доски. Определить может ли ферзь переместиться с первого поля на второе. Ферзь может перемещаться по горизонтали, по вертикали или диагонали.
16. Даны четыре числа – координаты двух полей шахматной доски. Определить может ли конь переместиться с первого поля на второе. Конь может перемещаться «буквой Г» – две клетки по горизонтали, одна по вертикали или две клетки по вертикали, одна по горизонтали.
17. Даны три различных числа. Расположите их в порядке возрастания.
18. Даны четыре числа. Найдите наибольшее из них.
19. Даны четыре различных числа. Расположите их в порядке убывания.