

Учреждение образования
«Белорусский государственный педагогический университет
имени Максима Танка»

Факультет физико-математический
Кафедра информатики и методики преподавания информатики

(рег. № 24-2-125 от 24.06.2016)
дата

СОГЛАСОВАНО

Заведующий кафедрой

С.И.Зенько
26 мая 2016 г.



СОГЛАСОВАНО

Декан факультета

С.И.Василец
26 мая 2016 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ПРАКТИКУМ ПО РЕШЕНИЮ ЗАДАЧ ПО ИНФОРМАТИКЕ
для специальности I – 02 05 02 Физика и информатика

Составитель:

С.В. Вабищевич, кандидат педагогических наук, доцент

Рассмотрено и утверждено

на заседании Совета БГПУ 24 июня 2016 г. протокол № 10

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ	7
РАЗДЕЛ 1. ПРОЦЕСС РЕШЕНИЯ ЗАДАЧ С ПОМОЩЬЮ КОМПЬЮТЕРА 7	
Тема 1.1. Этапы решения задач с помощью компьютера	7
РАЗДЕЛ 2. МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ.....	20
Тема 2.1. Общая характеристика методов и способов решения задач по информатике.....	20
Тема 2.2. Задачи обработки чисел.....	25
Тема 2.3. Поиск и сортировка.....	30
Тема 2.4. Задачи комбинаторики. Графы.....	35
ПРАКТИЧЕСКИЙ РАЗДЕЛ	38
РАЗДЕЛ 2. МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ.....	38
Тема 2.1. Общая характеристика методов и способов решения задач по информатике.....	38
Тема 2.2. Задачи обработки чисел.....	40
Тема 2.3. Поиск и сортировка.....	42
Тема 2.4. Задачи комбинаторики. Графы.....	43
РАЗДЕЛ 3. РЕШЕНИЕ ПРАКТИКООРИЕНТИРОВАННЫХ ЗАДАЧ. ТВОРЧЕСКИЙ ПРОЕКТ: ПОСТАНОВКА ЗАДАЧИ, ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ	44
Тема 3.1. Творческий проект: постановка практикоориентированной задачи	44
Тема 3.2. Использование современных технологий и средств программирования для обработки и визуализации данных	45
Тема 3.3. Творческий проект: верификация данных, защита	47
РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ.....	48
ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ.....	49

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Качество профессионально-методической подготовки будущего учителя информатики в значительной степени определяется уровнем профессиональной компетентности в сфере информатики. Одним из параметров, который определяет это является уровень форсированности обобщенных умений решать задачи из цикла специальных дисциплин специальностей «Математика и информатика», «Физика и информатика».

Систематизация содержания основных направлений информатики как фундаментальной науки, так и интенсивно развивающегося школьного учебного предмета – одна из главных задач учебной дисциплины **«Практикум по решению задач по информатике»**. Возможности и опыт, приобретаемый студентами в процессе изучения данной учебной дисциплины, позволят им осуществлять междисциплинарные практикоориентированные связи, поддерживать идеи профильного обучения в школе, развивать олимпиадное движение и исследовательскую деятельность по информатике, а также оптимизировать организационно-управленческий потенциал учебного процесса в системе образования.

Цель учебной дисциплины – формирование у будущих учителей информатики профессиональных компетенций для решения содержательно-деятельностных задач учебного процесса, обобщение знаний методов и современных подходов решения задач по информатике.

Основные задачи учебной дисциплины:

- формирование целостного системного представления о структуре процесса решения задач по информатике;
- обобщение знаний методов и способов решения задач по информатике;
- применение оптимальных методов и современных технологий к решению профессиональных задач;
- эффективная реализация творческого подхода при определении методов и средств решению задач информатики;
- мотивация к осуществлению самообразования и совершенствования в условиях интенсивного развития IT-сферы.

Место учебной дисциплины в системе подготовки специалиста

Изучение учебной дисциплины **«Практикум по решению задач по информатике»** опирается на основные академические, социально-личностные и профессиональные компетенции, сформированные у студентов в процессе изучения ими таких учебных дисциплин как «Педагогика», «Психология», «Технологии программирования и методы алгоритмизации», «Вычислительные методы и компьютерное моделирование», «Методика преподавания информатики». Благодаря ее изучению формируется целостное представление о методах и современных подходах к решению задач по информатике.

Профессиональные компетенции студента

Учебная дисциплина «Практикум по решению задач по информатике» входит в компонент учреждения высшего образования. Изучение учебной дисциплины «Практикум по решению задач по информатике» должно способствовать формированию у студентов академических, социально-личностных и профессиональных компетенций.

Требования к академическим компетенциям

Специалист должен:

- АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.
- АК-Ч. Уметь работать самостоятельно.
- АК-5. Быть способным порождать новые идеи (обладать креативностью).
- АК-6. Владеть междисциплинарным подходом при решении проблем.
- АК-10. Уметь осуществлять учебно-исследовательскую деятельность.

Требование к социально-личностным компетенциям

Специалист должен:

- СЛК-4. Владеть навыками здоровьесбережения.
- СЛК-6. Уметь работать в команде.
- СЛК-7. Быть способным к осуществлению самообразования и самосовершенствования профессиональной деятельности.

Требования к профессиональным компетенциям

Специалист должен быть способен:

Обучающая деятельность

- ПК-1. Управлять учебно-познавательной и учебно-исследовательской деятельностью обучающихся.
- ПК-2. Использовать оптимальные методы, формы, средства обучения.
- ПК-Ч. Организовывать самостоятельную работу обучающихся.

Воспитательная деятельность

- ПК-В. Формировать базовые компоненты культуры личности обучающегося.

Развивающая деятельность

- ПК-121 (ПК-1 12). Развивать навыки самостоятельной работы обучающихся с учебной, справочной, научной литературой и др. источниками информации.

Ценностно-ориентационная деятельность

- ПК-17. Осуществлять профессиональное самообразование и самовоспитание с целью совершенствования профессиональной деятельности.

В результате изучения учебной дисциплины студент должен знать:

- структуру процесса решения задач с помощью компьютера;
- методы и способы решения задач по информатике.

В результате изучения учебной дисциплины студент должен уметь:

- проводить системный анализ задачи, представленной на естественном языке, с учетом конкретной области деятельности человека;
- выбирать оптимальные методы и средства решения задачи;
- реализовывать все этапы решения задачи на компьютере;
- осуществлять верификацию полученных результатов.

В результате изучения учебной дисциплины студент должен владеть:

- методами поиска, анализа и дидактической адаптации научной информации по решению задач по информатике;
- современными технологиями и средствами для решения профессиональных задач.

Структура и содержание учебной дисциплины Учебная дисциплина **«Практикум по решению задач по информатике»** изучается на протяжении одного семестра и содержит три раздела. В первом разделе осуществляется теоретическая классификация видов задач и целостное, системное представление о структуре процесса решения задач по информатике. Во втором разделе рассматриваются методы и способы решения задач по информатике. Третий раздел посвящен реализации творческого проекта, предполагающего использование современных технологий и средств программирования при решении практикоориентированных задач и визуализации данных.

Данная учебная программа является основным документом, определяющим объем и содержание учебной дисциплины **«Практикум по решению задач по информатике»** для специальности 1-02 05 02 Физика и информатика.

Методы обучения

Обучение учебной дисциплине проходит в рамках организации лекционных и лабораторных занятий. При чтении лекций особое внимание следует уделять использованию мультимедийных технологий.

Организация лабораторных занятий предполагает использование лично-ориентированных методов обучения, что способствует развитию индивидуально-творческих способностей студентов и приобретению умений самостоятельной работы. Лабораторные работы направлены на формирование навыков решения профессионально-методических задач учителя информатики.

Содержание и формы самостоятельной работы студентов разрабатываются в соответствии с целями и задачами подготовки специалиста. Среди видов самостоятельной работы студентов представляется возможным применять: самостоятельную работу во время основных

аудиторных занятий (лекций, лабораторных занятий); самостоятельную работу в форме консультаций; внеаудиторную самостоятельную работу при выполнении студентами домашних заданий учебного и творческого характера. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

Распределение общего количества часов по формам обучения и семестрам

Специальность 1-02 05 02 Физика и информатика

Дневная форма получения высшего образования:

Всего на учебную дисциплину - 86 часов.

6 семестр — 42 часа аудиторных (8 часов — лекции, 34 часа - лабораторные занятия), 44 часа — самостоятельная работа.

Форма контроля — зачет (6 семестр).

РЕПОЗИТОРИЙ БГПУ

ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

РАЗДЕЛ 1. ПРОЦЕСС РЕШЕНИЯ ЗАДАЧ С ПОМОЩЬЮ КОМПЬЮТЕРА

Тема 1.1. Этапы решения задач с помощью компьютера

Особенности решения задач с помощью компьютера

Целью программирования является описание процессов обработки данных (в дальнейшем – просто процессов). Согласно требованиям международной федерация по обработке информации: **данные** – это представление фактов и идей в формализованном виде, пригодном для передачи и переработке в некоем процессе, а **информация** – это смысл, который придается данным при их представлении. Обработка данных – это выполнение систематической последовательности действий с данными. Данные представляются и хранятся на т.н. носителях данных. Совокупность носителей данных, используемых при какой-либо обработке данных, будем называть информационной средой. Набор данных, содержащихся в какой-либо момент в информационной среде, будем называть состоянием этой информационной среды. Процесс можно определить как последовательность сменяющих друг друга состояний некоторой информационной среды.

Описать процесс – значит определить последовательность состояний заданной информационной среды. Если мы хотим, чтобы по заданному описанию требуемый процесс порождался автоматически на каком-либо компьютере, необходимо, чтобы это описание было формализованным. Такое описание называется программой. С другой стороны, программа должна быть понятной и человеку, так как и при разработке программ, и при их использовании часто приходится выяснять, какой именно процесс она порождает. Поэтому программа составляется на удобном для человека формализованном языке программирования, с которого она автоматически переводится на язык соответствующего компьютера с помощью другой программы, называемой транслятором. Человеку (программисту), прежде чем составить программу на удобном для него языке программирования, приходится проделывать большую подготовительную работу по уточнению постановки задачи, выбору метода ее решения, выяснению специфики применения требуемой программы, прояснению общей организации разрабатываемой программы и многое другое. Использование этой информации может существенно упростить задачу понимания программы человеком, поэтому весьма полезно ее как-то фиксировать в виде отдельных документов (часто не формализованных, рассчитанных только для восприятия человеком).

Обычно программы разрабатываются в расчете на то, чтобы ими могли пользоваться люди, не участвующие в их разработке (их называют *пользователями*). Для освоения программы пользователем помимо ее текста

требуется определенная дополнительная документация. Программа или логически связанная совокупность программ на носителях данных, снабженная программной документацией, называется *программным средством (ПС)*. Программа позволяет осуществлять некоторую автоматическую обработку данных на компьютере. Программная документация позволяет понять, какие функции выполняет та или иная программа, как подготовить исходные данные и запустить требуемую программу в процесс ее выполнения, а также: что означают получаемые результаты (или каков эффект выполнения этой программы). Кроме того, программная документация помогает разобраться в самой программе, что необходимо, например, при ее модификации.

Каждое ПС должно выполнять определенные функции, т.е. делать то, что задумано. Хорошее ПС должно обладать еще целым рядом свойств, позволяющим успешно его использовать в течении длительного периода, т.е. обладать определенным качеством. *Качество ПС* – это совокупность его черт и характеристик, которые влияют на его способность удовлетворять заданные потребности пользователей. Это не означает, что разные ПС должны обладать одной и той же совокупностью таких свойств в их высшей возможной степени. Этому препятствует тот факт, что повышение качества ПС по одному из таких свойств часто может быть достигнуто лишь ценой изменения стоимости, сроков завершения разработки и снижения качества этого ПС по другим его свойствам. Качество ПС является удовлетворительным, когда оно обладает указанными свойствами в такой степени, чтобы гарантировать успешное его использование.

Совокупность свойств ПС, которая образует удовлетворительное для пользователя качество ПС, зависит от условий и характера эксплуатации этого ПС, т.е. от позиции, с которой должно рассматриваться качество этого ПС. Поэтому при описании качества ПС должны быть прежде всего фиксированы критерии отбора требуемых свойств ПС. В настоящее время критериями качества ПС принято считать:

- функциональность,
- надежность,
- легкость применения,
- эффективность,
- сопровождаемость,
- мобильность.

Функциональность – это способность ПС выполнять набор функций, удовлетворяющих заданным или подразумеваемым потребностям пользователей. Набор указанных функций определяется во внешнем описании ПС. Надежность подробно обсуждалась в первой лекции.

Легкость применения – это характеристики ПС, которые позволяют минимизировать усилия пользователя по подготовке исходных данных,

применению ПС и оценке полученных результатов, а также вызывать положительные эмоции определенного или подразумеваемого пользователя.

Эффективность – это отношение уровня услуг, предоставляемых ПС пользователю при заданных условиях, к объему используемых ресурсов.

Сопровождаемость – это характеристики ПС, которые позволяют минимизировать усилия по внесению изменений для устранения в нем ошибок и по его модификации в соответствии с изменяющимися потребностями пользователей.

Мобильность – это способность ПС быть перенесенным из одной среды (окружения) в другую, в частности, с одной ЭВМ на другую.

Функциональность и надежность являются обязательными критериями качества ПС, причем обеспечение надежности будет красной нитью проходить по всем этапам и процессам разработки ПС. Остальные критерии используются в зависимости от потребностей пользователей в соответствии с требованиями к ПС – их обеспечение будет обсуждаться в подходящих разделах курса.

Надежность является неотъемлемым атрибутом ПС. **Надежность ПС** – это его способность безотказно выполнять определенные функции при заданных условиях в течение заданного периода времени с достаточно большой вероятностью. При этом под отказом в ПС понимают проявление в нем ошибки. Надежное ПС не исключает наличия в нем ошибок - важно лишь, чтобы эти ошибки при практическом применении этого ПС в заданных условиях проявлялись достаточно редко. Убедиться, что ПС обладает таким свойством можно при его испытании путем тестирования, а также при практическом применении. Таким образом, фактически мы можем разрабатывать лишь надежные, а не правильные ПС.

Разрабатываемая ПС может обладать различной степенью надежности. Как измерять эту степень? Так же как в технике, степень надежности можно характеризовать вероятностью работы ПС без отказа в течении определенного периода времени. Однако в силу специфических особенностей ПС определение этой вероятности наталкивается на ряд трудностей по сравнению с решением этой задачи в технике.

При оценке степени надежности ПС следует также учитывать последствия каждого отказа. Некоторые ошибки в ПС могут вызывать лишь некоторые неудобства при его применении, тогда как другие ошибки могут иметь катастрофические последствия, например, угрожать человеческой жизни. Поэтому для оценки надежности ПС иногда используют дополнительные показатели, учитывающие стоимость (вред) для пользователя каждого отказа.

Рассмотрим теперь общие принципы обеспечения надежности ПС, что, как мы уже подчеркивали, является основным мотивом разработки ПС, задающим специфическую окраску всем технологическим процессам

разработки ПС. В технике известны четыре подхода обеспечению надежности:

- предупреждение ошибок;
- самообнаружение ошибок;
- самоисправление ошибок;
- обеспечение устойчивости к ошибкам.

Целью подхода предупреждения ошибок - не допустить ошибок в готовых продуктах, в нашем случае - в ПС. Проведенное рассмотрение природы ошибок при разработке ПС позволяет для достижения этой цели сконцентрировать внимание на следующих вопросах:

- борьбе со сложностью,
- обеспечении точности перевода,
- преодоления барьера между пользователем и разработчиком,
- обеспечения контроля принимаемых решений.

Этот подход связан с организацией процессов разработки ПС, т.е. с технологией программирования. И хотя, как мы уже отмечали, гарантировать отсутствие ошибок в ПС невозможно, но в рамках этого подхода можно достигнуть приемлемого уровня надежности ПС.

Остальные три подхода связаны с организацией самих продуктов технологии, в нашем случае - программ. Они учитывают возможность ошибки в программах. Самообнаружение ошибки в программе означает, что программа содержит средства обнаружения отказа в процессе ее выполнения. Самоисправление ошибки в программе означает не только обнаружение отказа в процессе ее выполнения, но и исправление последствий этого отказа, для чего в программе должны иметься соответствующие средства. Обеспечение устойчивости программы к ошибкам означает, что в программе содержатся средства, позволяющие локализовать область влияния отказа программы, либо уменьшить его неприятные последствия, а иногда предотвратить катастрофические последствия отказа. Однако, эти подходы используются весьма редко (может быть, относительно чаще используется обеспечение устойчивости к ошибкам). Связано это, во-первых, с тем, что многие простые методы, используемые в технике в рамках этих подходов, неприменимы в программировании, например, дублирование отдельных блоков и устройств (выполнение двух копий одной и той же программы всегда будет приводить к одинаковому эффекту - правильному или неправильному). А, во-вторых, добавление в программу дополнительных средств приводит к ее усложнению (иногда - значительному), что в какой-то мере мешает методам предупреждения ошибок.

Обеспечение качества программного средства

Общая характеристика процесса обеспечения качества программного средства.

Спецификация качества определяет основные ориентиры (цели), которые на всех этапах разработки ПС так или иначе влияют при принятии различных решений на выбор подходящего варианта. Однако, каждый примитив качества имеет свои особенности такого влияния, тем самым, обеспечения его наличия в ПС может потребовать своих подходов и методов разработки ПС или отдельных его частей. Кроме того, отмечалась также противоречивость критериев качества ПС и выражающих их примитивов качества: хорошее обеспечение одного какого-либо примитива качества ПС может существенно затруднить или сделать невозможным обеспечение некоторых других из этих примитивов. Поэтому существенная часть процесса обеспечения качества ПС состоит из поиска приемлемых компромиссов. Эти компромиссы частично должны быть определены уже в спецификации качества ПС: модель качества ПС должна конкретизировать требуемую степень присутствия в ПС каждого его примитива качества и определять приоритеты достижения этих степеней.

Обеспечение качества осуществляется в каждом технологическом процессе: принятые в нем решения в той или иной степени оказывают влияние на качество ПС в целом. В частности и потому, что значительная часть примитивов качества связана не столько со свойствами программ, входящих в ПС, сколько со свойствами документации. В силу отмеченной противоречивости примитивов качества весьма важно придерживаться выбранных приоритетов в их обеспечении. Но во всяком случае полезно придерживаться двух общих принципов:

- сначала необходимо обеспечить требуемую функциональность и надежность ПС, а затем уже доводить остальные критерии качества до приемлемого уровня их присутствия в ПС;
- нет никакой необходимости и может быть даже вредно добиваться более высокого уровня присутствия в ПС какого-либо примитива качества, чем тот, который определен в спецификации качества ПС.

Обеспечение функциональности и надежности ПС было рассмотрено в предыдущей лекции. Ниже обсуждается обеспечение других критериев качества ПС.

Обеспечение легкости применения программного средства.

Документированность ПС определяет состав пользовательской документации.

В предыдущей лекции было уже рассмотрено обеспечение двух из пяти примитивов качества (устойчивость и защищенность), которые определяют легкость применения ПС.

Документированность и информативность определяют состав и качество пользовательской документации.

Коммуникабельность обеспечивается созданием подходящего пользовательского интерфейса и соответствующей реализации исключительных ситуаций.

Обеспечение эффективности программного средства

Эффективность ПС обеспечивается принятием подходящих решений на разных этапах его разработки, начиная с разработки его архитектуры. Особенно сильно на эффективность ПС (особенно по памяти) влияет выбор структуры и представления данных. Но и выбор алгоритмов, используемых в тех или иных программных модулях, а также особенности их реализации (включая выбор языка программирования) может существенно повлиять на эффективность ПС. При этом постоянно приходится разрешать противоречие между временной эффективностью и эффективностью по памяти. Поэтому весьма важно, чтобы в спецификации качества было явно указано количественное соотношение между показателями этих примитивов качества или хотя бы заданы количественные границы для одного из этих показателей. И все же разные программные модули по-разному влияют на эффективность ПС в целом: и по вкладу в общие затраты ПС по времени и памяти, и по влиянию на разные примитивы качества (одни модули могут сильно влиять на достижение временной эффективности и практически не влиять на эффективность по памяти, а другие могут существенно влиять на общий расход памяти, не оказывая заметного влияния на время работы ПС). Более того, это влияние (прежде всего в отношении временной эффективности) заранее (до окончания реализации ПС) далеко не всегда можно правильно оценить.

С учетом сказанного, рекомендуется придерживаться следующих принципов для обеспечения эффективности ПС:

- сначала нужно разработать надежное ПС, а уж потом добиваться требуемой его эффективности в соответствии со спецификацией качества этого ПС;
- для повышения эффективности ПС используйте, прежде всего, оптимизирующий компилятор - это может обеспечить требуемую эффективность;
- если достигнутая эффективность ПС не удовлетворяет спецификации его качества, то найдите самые критические модули с точки зрения требуемой эффективности ПС (в случае временной эффективности для этого потребуется получить распределение по модулям времени работы ПС путем соответствующих измерений во время выполнения ПС); эти модули и попытайтесь оптимизировать в первую очередь путем их ручной переделки;
- не занимайтесь оптимизацией модуля, если этого не требуется для достижения требуемой эффективности ПС.

Обеспечение сопровождаемости.

Информативность и понятность определяют состав и качество документации по сопровождению. Кроме того, относительно текстов программ (модулей) можно сделать следующие рекомендации.

- используйте в тексте модуля комментарии, проясняющие и объясняющие особенности принимаемых решений; по-возможности, включайте комментарии (хотя бы в краткой форме) на самой ранней стадии разработки текста модуля;

- используйте осмысленные (мнемонические) и устойчиво различимые имена (оптимальная длина имени - 4-12 литер, цифры - в конце), не используйте сходные имена и ключевые слова;

- соблюдайте осторожность в использовании констант (уникальная константа должна иметь единственное вхождение в текст модуля: при ее объявлении или, в крайнем случае, при инициализации переменной в качестве константы);

- не бойтесь использовать не обязательные скобки (скобки обходятся дешевле, чем ошибки);

- размещайте не больше одного оператора в строке; для прояснения структуры модуля используйте дополнительные пробелы (отступы) в начале каждой строки;

- избегайте трюков, т.е. таких приемов программирования, когда создаются фрагменты модуля, основной эффект которых не очевиден или скрыт (завуалирован), например, побочные эффекты функций.

Структурированность и модульность упрощают как понимание текстов программ, так и их модификацию.

Аттестация ПС – это авторитетное подтверждение качества ПС. Обычно для аттестации ПС создается представительная (аттестационная) комиссия из экспертов, представителей заказчика и представителей разработчика. Эта комиссия проводит испытания ПС с целью получения необходимой информации для оценки его качества. Под испытанием ПС мы будем понимать процесс проведения комплекса мероприятий, исследующих пригодность ПС для успешной его эксплуатации (применения и сопровождения) в соответствии с требованиями заказчика. Этот комплекс включает проверку полноты и точности программной документации, изучение и обсуждение других ее свойств, а также необходимое тестирование программ, входящих в состав ПС, и, в частности, соответствия этих программ имеющейся документации.

На основе информации, полученной во время испытаний ПС, прежде всего должно быть установлено, что ПС выполняет декларируемые функции, а также должно быть установлено, в какой степени ПС обладает декларируемыми примитивами и критериями качества. Таким образом, оценка качества ПС является основным содержанием процесса аттестации. Произведенная оценка качества ПС фиксируется в соответствующем решении аттестационной комиссии.

Виды испытаний программного средства.

Известны следующие виды испытаний ПС, проводимых с целью аттестации ПС:

- испытания компонент ПС;
- системные испытания;
- приемо-сдаточные испытания;
- полевые испытания;
- промышленные испытания.

Испытания компонент ПС - это проверка (тестирование) работоспособности отдельных подсистем ПС. Проводятся только в исключительных случаях по специальному решению аттестационной комиссии.

Системные испытания ПС - это проверка (тестирование) работоспособности ПС в целом. Может включать те же виды тестирования, что и при комплексной отладке ПС. Проверка проводится по решению аттестационной комиссии, если возникают сомнения в качестве проведения отладки разработчиками ПС.

Приемо-сдаточные испытания являются основным видом испытаний при аттестации ПС. Именно с этих испытаний начинает работу аттестационная комиссия. Эти испытания начинаются с изучения представленной документации, в том числе, и документации по тестированию и отладке ПС. Если в документации отсутствуют достаточно полные результаты тестирования ПС, аттестационная комиссия может принять решение о проведении системных испытаний ПС или о прекращении процесса аттестации с рекомендацией разработчику провести дополнительное (более полное) тестирование ПС. Кроме того, во время этих испытаний могут выборочно пропускаться тесты разработчиков, а также контрольные задачи пользователей и дополнительные тесты, подготовленные комиссией для оценки качества аттестуемого ПС.

Полевые испытания ПС - это демонстрация ПС вместе с технической системой, которой управляет эта ПС, узкому кругу заказчиков в реальных условиях и осуществляется тщательное наблюдение за поведением ПС. Заказчикам должна быть предоставлена возможность задания собственных контрольных примеров, в частности, с выходов в критические режимы работы технической системы, а также с вызовом в ней аварийных ситуаций. Это дополнительные испытания, проводимые по решению аттестационной комиссии только для некоторых ПС, управляющих определенными техническими системами.

Промышленные испытания ПС - это процесс передачи ПС в постоянную эксплуатацию пользователям. Представляет собой период опытной эксплуатации ПС пользователями со сбором информации об особенностях поведения ПС и ее эксплуатационных характеристиках. Это завершающие испытания ПС, которые проводятся по решению

аттестационной комиссии, если на предшествующих испытаниях получена недостаточно полная или надежная информация для оценки качества аттестуемого ПС.

Методы оценки качества программного средства.

Оценка качества ПС по каждому из критериев сводится к оценке каждого из примитивов, связанных с этим критерием качества ПС, в соответствии с их конкретизацией, произведенной в спецификации качества этого ПС. Методы оценки примитивов качества ПС можно разделить на четыре группы:

- непосредственное измерение показателей примитива качества;
- обработка программ и документации ПС специальными программными инструментами (процессорами);
- тестирование программ ПС;
- экспертная оценка на основании изучения программ и документации ПС.

Непосредственное измерение показателей примитива качества производится путем подсчета числа вхождений в тот или иной программный документ характерных единиц, объектов, конструкций и т.п., а также путем измерения времени работы различных устройств и объема занятой памяти ЭВМ при выполнении контрольных примеров. Например, некоторым показателем эффективности по памяти может быть число строк программы на языке программирования, а некоторым показателем эффективности по времени может быть время ответа на запрос. Использование каких-либо показателей для примитивов качества может определяться в спецификации качества ПС. Метод непосредственного измерения показателей примитива качества может сочетаться с использованием тестирования программ.

Для установления наличия у ПС некоторых примитивов качества могут использоваться определенные программные инструментальные средства. Такие программные инструменты обрабатывают тексты программ или программной документации с целью контроля каких-либо примитивов качества или получения некоторых показателей этих примитивов качества. Для оценки структурированности программ ПС, если они программировались на подходящем структурном диалекте базового языка программирования, достаточно было бы их пропустить через конвертер структурированных программ, осуществляющий синтаксический и некоторый семантический контроль этого диалекта и переводящий тексты этих программ на входной язык базового транслятора. Однако таким путем в настоящее время удастся контролировать лишь небольшое число примитивов качества, да и то в редких случаях. В ряде случаев вместо программных инструментов, контролирующих качество ПС, полезнее применять инструменты, осуществляющие преобразование представления программ или программной документации. Таким, например, является форматор программ, приводящий тексты программ к удобочитаемому виду, - обработка текстов

программ ПС таким инструментом может автоматически обеспечить наличие соответствующего примитива качества у ПС.

Для оценки некоторых примитивов качества ПС используется тестирование. К таким примитивам относится прежде всего завершенность ПС, а также его точность, устойчивость, защищенность и другие примитивы качества. В ряде случаев для оценки отдельных примитивов качества ПС тестирование применяется в сочетании с другими методами. Так для оценки качества документации по применению ПС (П-документированности) тестирование применяется в сочетании с экспертной оценкой этой документации. Если при комплексной отладке ПС было проведено достаточно полное тестирование, то эти же тесты могут быть использованы и при аттестации ПС. В этом случае аттестационная комиссия может воспользоваться протоколами тестирования, проведенного при комплексной отладке. Однако и в этом случае необходимо выполнить какие-либо новые тесты или хотя бы повторно некоторые старые. Если же тестирование при комплексной отладке будет признано недостаточно полным, то необходимо провести более полное тестирование. В этом случае может быть принято решение о проведении испытаний компонент или системных испытаний ПС, а также о возврате ПС разработчикам на доработку. Весьма важно, чтобы для оценки ПС по критерию легкости применения было проведено (во время отладки и аттестации ПС) полное тестирование по тестам, подготовленным на основании документации по применению, а по критерию сопровождаемости - по тестам, подготовленным по каждому из документов, предлагаемых для сопровождения ПС.

Для оценки большинства примитивов качества ПС в настоящее время можно применять только метод экспертных оценок. Этот метод заключается в следующем: назначается группа экспертов, каждый из этих экспертов в результате изучения представленной документации составляет свое мнение об обладании ПС требуемым примитивом качества, а затем голосованием членов этой группы устанавливается оценка требуемого примитива качества ПС. Эта оценка может производиться как по двухбалльной системе ("обладает" - "не обладает"), так и учитывать степень обладания ПС этим примитивом качества (например, производиться по пятибалльной системе).

Отладка программного средства.

Отладка ПС - это деятельность, направленная на обнаружение и исправление ошибок в ПС с использованием процессов выполнения его программ.

Тестирование ПС - это процесс выполнения его программ на некотором наборе данных, для которого заранее известен результат применения или известны правила поведения этих программ. Указанный набор данных называется тестовым или просто тестом. Таким образом, отладку можно представить в виде многократного повторения трех процессов: тестирования, в результате которого может быть констатировано

наличие в ПС ошибки, поиска места ошибки в программах и документации ПС и редактирования программ и документации с целью устранения обнаруженной ошибки. Другими словами:

Отладка = Тестирование + Поиск ошибок + Редактирование.

В зарубежной литературе отладку часто понимают только как процесс поиска и исправления ошибок (без тестирования), факт наличия которых устанавливается при тестировании. Иногда тестирование и отладку считают синонимами. В нашей стране в понятие отладки обычно включают и тестирование, поэтому мы будем следовать сложившейся традиции.

Принципы и виды отладки.

Успех отладки в значительной степени предопределяет рациональная организация тестирования. При отладке отыскиваются и устраняются, в основном, те ошибки, наличие которых в ПС устанавливается при тестировании. Как было уже отмечено, тестирование не может доказать правильность ПС, в лучшем случае оно может продемонстрировать наличие в нем ошибки. Другими словами, нельзя гарантировать, что тестированием ПС практически выполнимым набором тестов можно установить наличие каждой имеющейся в ПС ошибки. Поэтому возникает две задачи. Первая: подготовить такой набор тестов и применить к ним ПС, чтобы обнаружить в нем по возможности большее число ошибок. Однако чем дольше продолжается процесс тестирования (и отладки в целом), тем большей становится стоимость ПС. Отсюда вторая задача: определить момент окончания отладки ПС (или отдельной его компоненты). Признаком возможности окончания отладки является полнота охвата пропущенными через ПС тестами (т.е. тестами, к которым применено ПС) множества различных ситуаций, возникающих при выполнении программ ПС, и относительно редкое проявление ошибок в ПС на последнем отрезке процесса тестирования. Последнее определяется в соответствии с требуемой степенью надежности ПС, указанной в спецификации его качества.

Для оптимизации набора тестов, т.е. для подготовки такого набора тестов, который позволял бы при заданном их числе (или при заданном интервале времени, отведенном на тестирование) обнаруживать большее число ошибок, необходимо, во-первых, заранее планировать этот набор и, во-вторых, использовать рациональную стратегию планирования (проектирования) тестов.

Определяются разные виды тестирования в зависимости от вида программного документа, на основании которого строятся тесты. В нашей стране различаются два основных вида отладки (включая тестирование): автономную и комплексную отладку. Автономная отладка означает тестирование только какой-то части программы, входящей в ПС, с поиском и исправлением в ней фиксируемых при тестировании ошибок. Она фактически включает отладку каждого модуля и отладку сопряжения модулей. Комплексная отладка означает тестирование ПС в целом с поиском

и исправлением фиксируемых при тестировании ошибок во всех документах (включая тексты программ ПС), относящихся к ПС в целом. К таким документам относятся определение требований к ПС, спецификация качества ПС, функциональная спецификация ПС, описание архитектуры ПС и тексты программ ПС.

Заповеди отладки.

В данном разделе даются общие рекомендации по организации отладки. Но сначала следует отметить некоторый феномен, который подтверждает важность предупреждения ошибок на предыдущих этапах разработки: по мере роста числа обнаруженных и исправленных ошибок в ПС растет также относительная вероятность существования в нем необнаруженных ошибок. Это объясняется тем, что при росте числа ошибок, обнаруженных в ПС, уточняется и наше представление об общем числе допущенных в нем ошибок, а значит, в какой-то мере, и о числе необнаруженных еще ошибок. Этот феномен подтверждает важность раннего обнаружения ошибок и необходимость тщательного контроля принимаемых решений на каждом этапе разработки ПС. Ниже приводятся рекомендации по организации отладки в форме заповедей.

Заповедь 1. Считайте тестирование ключевой задачей разработки ПС, поручайте его самым квалифицированным и одаренным программистам; нежелательно тестировать свою собственную программу.

Заповедь 2. Хорош тот тест, для которого высока вероятность обнаружить ошибку, а не тот, который демонстрирует правильную работу программы.

Заповедь 3. Готовьте тесты как для правильных, так и для неправильных данных.

Заповедь 4. Избегайте невоспроизводимых тестов, документируйте их пропуск через компьютер; детально изучайте результаты каждого теста.

Заповедь 5. Каждый модуль подключайте к программе только один раз; никогда не изменяйте программу, чтобы облегчить ее тестирование.

Заповедь 6. Пропускайте заново все тесты, связанные с проверкой работы какой-либо программы ПС или ее взаимодействия с другими программами, если в нее были внесены изменения (например, в результате устранения ошибки).

При автономной отладке каждый модуль на самом деле тестируется в некотором программном окружении, кроме случая, когда отлаживаемая программа состоит только из одного модуля. Это окружение состоит из других модулей, часть которых является модулями отлаживаемой программы, которые уже отлажены, а часть - модулями, управляющими отладкой (отладочными модулями, см. ниже). Таким образом, при автономной отладке тестируется всегда некоторая программа, построенная специально для тестирования отлаживаемого модуля.

Комплексная отладка программного средства.

При комплексной отладке тестируется ПС в целом, причем тесты готовятся по каждому из документов ПС. Тестирование этих документов производится, как правило, в порядке, обратном их разработке (исключение составляет лишь тестирование документации по применению, которая разрабатывается по внешнему описанию параллельно с разработкой текстов программ; это тестирование лучше производить после завершения тестирования внешнего описания). Тестирование при комплексной отладке представляет собой применение ПС к конкретным данным, которые в принципе могут возникнуть у пользователя (в частности, все тесты готовятся в форме, рассчитанной на пользователя), но, возможно, в моделируемой (а не в реальной) среде. Например, некоторые недоступные при комплексной отладке устройства ввода и вывода могут быть заменены их программными имитаторами.

РЕПОЗИТОРИЙ БГПУ

РАЗДЕЛ 2. МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ

Тема 2.1. Общая характеристика методов и способов решения задач по информатике

Педагоги давно пришли к выводу, что на современном уроке «передача знаний» не является главной целью. Более того, такая организация урока, где детям выдаются готовые рафинированные знания, губительна для познавательной деятельности. Ученик должен быть поставлен в условия поиска, заблуждений, радостных открытий и огорчений от временных неудач. Мышление начинается там, где есть неизвестное, где встречаются затруднение, непонимание, ошибка. А, следовательно, учитель так продумывает урок, что поставленные учебные цели подразделяются на ряд учебных задач, решаемых учащимися совместно или самостоятельно. Еще великий Сократ отмечал, что «учитель – не тот, кто дает, а тот, у кого берут». Система запланированных целей неразрывно связана с системой действий, которые ведут к выполнению этих целей. Тем, что вызывает эти активные действия, становятся учебные задачи, выступающие как разновидность опережающего управления познавательной деятельностью, «проект будущего учебного действия», определяющий интеллектуальное пространство, в котором ученик станет выполнять мыслительные операции. Несмотря на большие изменения в российском образовании, по-прежнему наиболее традиционным является «знаниевый» подход в обучении.

Суть его заключается в том, чтобы передать учащимся необходимый объем знаний. Единица обучения – порция знаний, т. е. предметной информации, в которой ученики должны осознать значение и смысл. «Знаниевый» подход развивает почти исключительно интеллектуальную сферу сознания, причем только ту ее часть, которая связана с памятью, совершенно не затрагивая волевою, эмоционально-чувственную и мотивационную сферы сознания обучаемых. Преобладание «знаниевой» подготовки над всеми остальными видами обучения приводит к развитию формального уровня образования. Такая подготовка затрудняет переход от обучения в образовательном учреждении к многообразной жизнедеятельности выпускника школы.

Относительно недавно сформировался «задачный» подход, который основной акцент делает на разрешение в ходе обучения различных учебных задач, вопросов, ситуаций и т. д. Единица такого обучения – интеллектуальное умение (или даже навык), позволяющее разрешать учебные задачи, давать ответы на вопросы. «Задачный» подход интенсивно развивает интеллектуальную сферу сознания, но в отличие от «знаниевого» – прежде всего, логическое мышление. Специально организованное, правильно и систематически осуществляемое обучение в виде разрешения разнообразных учебных задач расширяет возможности обучения.

Развивается ориентировочная сторона учебной деятельности, когда ученик активно занимается поиском правильного решения, самостоятельно добывает новые знания. Кроме того, возрастает роль аналитико-поисковой деятельности по определению последствий учебных действий, сокращается набор «проб» и «ошибок», появляется стремление найти все возможные для данной задачи решения, возрастает вариативность действий. При этом существенно развиваются волевая и мотивационная сферы учебной деятельности и в некоторой степени эмоционально-чувственная. Имеющиеся таксономии учебных целей, и особенно таксономия Б. Блума, широко используются в образовательных технологиях при построении системы типовых задач для определения конкретных умений учащихся. М.Е. Бершадский и М.Е. Гузеев приводят конкретный пример заданий для школьников из различных предметных областей, по результатам которых можно судить о различных уровнях обученности. Примеры задач для различных уровней учебных целей в познавательной области по Бенджамину Блуму. Знание. Дайте определение имени существительного. Когда и где разворачивается действие рассказа и когда в тексте это становится ясно? На какие две категории можно разделить все живые существа? Вычислите значение выражения $(9-177)/21+8$. Назовите палаты Федерального Собрания Российской Федерации. Понимание. Расскажите, что произошло с героями романа после эпизода, изображенного на иллюстрации. Нарисуйте три картинку о событиях, которые происходили в начале, середине, и конце XIX века в Германии.

Расскажите коротко своими словами содержание параграфа. Напишите сочинение (две-три страницы) с описанием обязанностей президента. Приведите примеры того, как в рассказе подтверждается, что Даниле не безразличны чувства других людей. Почему на географических картах используются символы? Применение. Составьте предложение с использованием каждого из данных слов. Изучив знаки на картах, составьте план маршрута для посещения трех исторических памятников в Подмосковье. Найдите площадь класса, в котором вы занимаетесь. Теперь, когда вы прочли почти весь рассказ, угадайте, чем он закончится. Придумайте три практических ситуации, в которых вы сможете использовать изученный способ определения влажности. Анализ. Какие события в этой истории действительно имели место, а какие выдуманы автором?

Чем период после гражданской войны напоминает период после войны в Афганистане, и чем они отличаются? Распределите минералы этой коллекции по трем категориям. Среди утверждений разбираемой статьи укажите три, которые являются фактами, и три, которые являются мнениями. По описаниям десяти насекомых определите, к какому семейству относится каждое из них. Какие из данных слов имеют латинское происхождение, а какие – греческое? Синтез. Выдумайте какую-нибудь страну и нарисуйте ее

карту, используя не менее 15 географических элементов. Представьте себе, что вы присутствуете на традиционной церемонии в Санкт-Петербурге.

Напишите дневник событий. Напишите свой рассказ «Му-Му» с позиции собаки. Теперь, когда вы видели результаты экспериментов, предложите гипотезу, которая бы объяснила эти результаты. Напишите данный отчет о событиях теперь в жанре фельетона. Предложите различные способы решения проблемы озеленения школьного двора. Оценка. Обязательно ли должны все кандидаты на пост мэра Екатеринбурга иметь высшее образование? Сравните двух главных героев поэмы. Нужно ли требовать от школьников навыков устных вычислений в век недорогих калькуляторов? Почему? Оправдано ли насилие для исправления несправедливости? Объясните вашу позицию. Какая среда из всех изученных вам больше нравится для жизни? Аргументируйте свой ответ. В каких случаях демократическое устройство государства менее эффективно, чем диктатура? (Б.Е. Бершадский, В.В. Гузеев. Дидактические и психологические основания образовательной технологии. М., 2003. С. 144-146.) Отнесение большинства описанных учебных задач к определенному уровню обученности не вызывает сомнения, что, по замечанию цитируемых авторов, «и является необходимым критерием применимости таксономии для разработки системы мониторинга». В то же время некоторые из заданий, к примеру, на применение можно было бы отнести к уровню анализа или даже оценки. Во всяком случае, создатели различных образовательных технологий, как правило, стремятся к разработке собственных таксономий учебных задач. При всем разнообразии подходов привлекает внимание стремление специалистов с помощью таксономии проследить весь учебный процесс, создать систему учебного мониторинга не только для определения достижений учащихся, но и динамики всего учебного процесса.

Одной из наиболее популярных является разработка известного чешского ученого Даны Толлингеровой, которая, опираясь на таксономию учебных целей Б. Блума, предложила таксономию учебных задач, разделенных на 5 категорий, содержащих 27 типов учебных задач по операционной структуре, то есть по операциям, необходимым для их выполнения. Полнота и детальная разработанность таксономии учебных задач, а также то обстоятельство, что задания в ней упорядочены по когнитивной сложности и сопоставлены с количественным показателем (тип задач), позволяет использовать эту таксономию не только как оценочную шкалу, но и для разработки алгоритма учебных действий.

ТАКСОНОМИЯ УЧЕБНЫХ ЗАДАЧ ПО Д. ТОЛЛИНГЕРОВОЙ

1. Задачи, требующие мнемического воспроизведения данных: Задачи по узнаванию. Задачи по воспроизведению отдельных фактов, чисел, понятий. Задачи по воспроизведению дефиниций, норм, правил. Задачи по воспроизведению больших текстов, блоков, стихов, таблиц и т.п.

2. Задачи, требующие простых мыслительных операций с данными: Задачи по выявлению фактов (измерение, взвешивание, простые исчисления и т.п.). Задачи по перечислению и описанию фактов (исчисление, перечень и т.п.). Задачи по перечислению и описанию процессов и способов действий. Задачи по разбору и структуре (анализ и синтез). Задачи по сопоставлению и различению (сравнение и разделение), Задачи по распределению (категоризация и классификация). Задачи по выявлению взаимоотношений между фактами (причина, следствие, цель, средство, влияние, функция, полезность, инструмент, способ и т.п.). Задачи по абстракции, конкретизации и обобщению. Решение несложных примеров (с неизвестными величинами и т.д.)

3. Задачи, требующие сложных мыслительных операций с данными: Задачи по переносу (трансляция, трансформация). Задачи по изложению (интерпретация, разъяснение смысла, значения, обоснование). Задачи по индукции. Задачи по дедукции. Задачи по доказыванию (аргументацией и проверке (верификации)) Задачи по оценке.

4. Задачи, требующие сообщения данных: Задачи по разработке обзоров, конспектов, содержания и т.д. Задачи по разработке отчетов, трактатов, докладов и т.п. Самостоятельные письменные работы, чертежи, проекты и т.п.

5. Задачи, требующие творческого мышления. Задачи по практическому приложению. Решение проблемных задач и ситуаций. Постановка вопросов и формулировка задач или заданий.

Задачи по обнаружению на основании собственных наблюдений (на сенсорной основе). Задачи по обнаружению на основании собственных размышлений (на рациональной основе) К первой категории относятся задачи, требующие от учащегося мнемических операций, содержание которых предусматривает узнавание операций, содержание которых предусматривает узнавание или репродукцию отдельных фактов или их целого. Чаще всего они начинаются со слов: какая из...; что это?; как называется...; кто был...; дайте дефиницию (определение)... и т.д. Во вторую категорию включены задачи, при решении которых уже необходимы элементарные операции.

Это задачи по выявлению, перечислению, сопоставлению, обобщению и т.п. Начинаются они обычно словами: установите... какого размера... опишите, из чего состоит... перечислите части... составьте перечень... опишите, как протекает... скажите, как проводится... как действуем при... чем отличается... сравните... определите сходства и различия... почему... каким способом... что является причиной... и т.п. Третья категория охватывает задачи, решение которых требует сложных мыслительных операций. Сюда относятся задачи по индукции, дедукции, интерпретации, верификации и др. Начинаются они обычно со слов: объясните смысл...;

раскройте значение...; как вы понимаете...; почему думаете, что...; определите...; докажите... и т.д.

Следует указать, что к этой категории относятся все задачи, в которых учащиеся должны перевести что-то с одного «языка» на другой, например, выразить формулу, прочесть что-либо, перевести текст с родного языка на иностранный и т.д. В четвертую категорию включены задачи, предусматривающие для их решения помимо мыслительных операций еще какой-нибудь речевой акт, устный или письменный. Следовательно, сюда относятся все задачи, требующие не только проведение определенных операций, но и высказываний о них. Учащийся в этих задачах дает показание не только о результате решения, но также и о его ходе, условиях, фазах, компонентах, трудностях и т.д. В пятую категорию входят задачи, которые предполагают самостоятельность при решении задач. Начинаются они обычно словами: придумай практический пример...; обрати внимание...; на основании собственных наблюдений определи... и т.п. Это уже задачи, которые предполагают не только знание всех предшествующих операций, но и способность комбинировать их в большие блоки, структуры, секвенции, стратегии и пр. так, чтобы они создавали нечто новое, пусть даже только субъектно, то есть для учащегося, новое. Дидактическая ценность системы учебных задач, по мнению Д.Толлингеровой, связана с выполнением поставленной педагогической цели: если целью учителя было проверить знания учащегося, то достаточно, чтобы тест содержал задачи первой категории, если же цель – проверить, как учащийся использует сложные мыслительные операции, то задачи 1-2 категорий не позволят гарантировать достижения поставленной цели.

При этом, сравнительно разнородный набор задач, где чередуются разные познавательные операции, предотвращает демотивацию учащихся под влиянием монотонности задаваемых задач. Как видим, все задачи, предложенные Д. Толлингеровой, проранжированы по возрастанию когнитивной сложности и операциональной ценности. Система запланированных целей неразрывно связана с системой действий, которые ведут к выполнению этих целей. Тем, что вызывает эти активные действия, становятся учебные задачи, выступающие как разновидность опережающего управления когнитивной деятельностью, «проект будущего учебного действия», определяющий интеллектуальное пространство, в котором ученик станет выполнять мыслительные действия.

Таким образом, с использованием таксономии можно конструировать систему учебных задач для выполнения поставленных педагогических целей, более полно учитывать состав когнитивных требований к учебной ситуации, проводить диагностику знаний и уровня сформированности учебных действий учащихся, а также прогнозировать ход обучения с учетом меры сложности задач и степени нагрузки на все виды проектируемой познавательной деятельности, то есть можно создавать индивидуальную

программу развития обучаемых или более эффективно организовывать дифференцированную работу. По мнению Д.Толлингеровой, главная задача учителя – научиться составлять учебные задачи так, чтобы их операционная структура соответствовала преследуемым педагогическим целям и учебному материалу.

Тема 2.2. Задачи обработки чисел

Позиционные и непозиционные системы счисления

Система счисления — символический метод записи чисел, представление чисел с помощью письменных знаков. Символы, при помощи которых записывается число, называются цифрами. Система счисления: дает представления множества чисел (целых или вещественных) дает каждому числу уникальное представление (или, по крайней мере, стандартное представление) отражает алгебраическую и арифметическую структуру чисел. Разные народы в разные времена использовали разные системы счисления. Следы древних систем счета встречаются и сегодня в культуре многих народов. К древнему Вавилону восходит деление часа на 60 минут и угла на 360 градусов. К Древнему Риму - традиция записывать в римской записи числа I, II, III и т. д. К англосаксам - счет дюжинами: в году 12 месяцев, в футе 12 дюймов, сутки делятся на 2 периода по 12 часов.

По современным данным, развитые системы нумерации впервые появились в древнем Египте. Для записи чисел египтяне применяли иероглифы один, десять, сто, тысяча и т.д. Все остальные числа записывались с помощью этих иероглифов и операции сложения. Недостатки этой системы - невозможность записи больших чисел и громоздкость. В конце концов, самой популярной системой счисления оказалась десятичная система.

Десятичная система счисления пришла из Индии, где она появилась не позднее VI в. н. э. В ней всего 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 но информацию несет не только цифра, но также и место позиция, на которой она стоит. В числе 444 три одинаковых цифры обозначают количество и единиц, и десятков, и сотен. А вот в числе 400 первая цифра обозначает число сотен, два 0 сами по себе вклад в число не дают, а нужны лишь для указания позиции цифры 4.

Классификация систем счисления

Системы счисления подразделяются на позиционные и непозиционные. Позиционные системы счисления Позиционные системы счисления (СС) - это системы счисления, в которых количественный эквивалент каждой цифры (вес) зависит от ее положения (позиции) в записи числа.

Путем долгого развития человечество пришло к созданию позиционного принципа записи чисел, который состоит в том, что каждая цифра, содержащаяся в записи числа, занимает определенное место, называемое разрядом. Отсчет разрядов производится справа налево. Единица

каждого следующего разряда всегда превосходит единицу предыдущего разряда в определенное число раз. Это отношение носит название основание системы счисления (у непозиционных систем счисления понятия «разряда» и «основания» отсутствуют). Например: число 237 состоит из 3 цифр. Понятно, что отдельно взятая цифра 7 больше чем цифра 2. Однако, в составе числа, двойка стоит на позиции сотен, а семерка - на позиции единиц, поэтому количественное представление двойки - две сотни, или двести, а семерка - все та же семь. Многие, кроме десятичной СС, о других позиционных системах не имеют представления, хотя и часто ими пользуются. Например: шестидесятеричная (Древний Вавилон) - первая позиционная система счисления.

До сих пор при измерении времени используется основание равное 60 (1 мин = 60 с, 1 ч = 60 мин); двенадцатеричная система счисления (широкое распространение получила в XIX в. Число 12 - «дюжина»: в сутках две дюжины часов. Счет не по пальцам, а по суставам пальцев. На каждом пальце руки, кроме большого, по 3 сустава - всего 12; В настоящее время наиболее распространенными позиционными системами счисления являются десятичная, двоичная, восьмеричная и шестнадцатеричная. Общее свойство всех позиционных систем счисления: при каждом переходе влево (вправо) в записи числа на один разряд величина цифры увеличивается (уменьшается) во столько раз, чему равно основание системы счисления.

Достоинства позиционных систем счисления: в позиционных системах счисления устранены все недостатки непозиционных: в них можно записать любое число (как натуральное, так и действительное); запись чисел компактна и удобна; благодаря поразрядной организации записи чисел с ними легко проводить математические операции.

Непозиционные системы счисления

В непозиционных системах счисления величина, которую обозначает цифра, не зависит от положения в числе. Например: Римская система счисления. Из многочисленных представителей этой группы в настоящее время сохранила свое значение лишь римская система счисления, где для обозначения цифр используются латинские буквы:

I V X L C D M 1 5 10 50 100 500 1000

С их помощью можно записывать натуральные числа. Например, число 1995 будет представлено, как MCMXCV (M-1000, CM-900, XC-90 и V-5). Правила записи чисел в римской системе счисления: если большая цифра стоит перед меньшей, они складываются, например: VI – 6 (5+1); если меньшая цифра стоит перед большей, то из большей вычитается меньшая, причем в этом случае меньшая цифра уже повторяться не может, например: XL — 40 (50-10), XXL – нельзя; цифры M, C, X, I могут повторяться в записи числа не более трех раз подряд; цифры D, L, V могут использоваться в записи числа только по одному разу. Например, запись XXX обозначает число 30, состоящее из трех цифр X, каждая из которых, независимо от места

ее положения в записи числа, равна 10. Запись MCXXIV обозначает 1124, а самое большое число, которое можно записать в этой системе счисления, это число MMMCMXCIX (3999). Для записи еще больших чисел пришлось бы вводить все новые обозначения. По этой причине, а также по причине отсутствия цифры ноль, римская система счисления не годится для записи действительных чисел. Таким образом, можно констатировать следующие основные недостатки непозиционных систем счисления: в них нельзя записать любое число; запись чисел обычно громоздка и неудобна; математические операции над ними крайне затруднены.

Алфавит и основание системы счисления

Алфавитом системы счисления называется совокупность различных цифр, используемых в позиционной системе счисления для записи чисел. Например:

Десятичная система: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

Двоичная система: $\{0, 1\}$

Восьмеричная система: $\{0, 1, 2, 3, 4, 5, 6, 7\}$

Шестнадцатеричная система: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$

Количество цифр в алфавите равно основанию системы счисления. Основанием позиционной системы счисления называется количество знаков или символов, используемых для изображения числа в данной системе счисления. Базисом позиционной системы счисления называется последовательность чисел, каждое из которых задает количественное значение или «вес» каждого разряда. Например: Базисы некоторых позиционных систем счисления.

Десятичная система: $100, 101, 102, 103, 104, \dots, 10n, \dots$

Двоичная система: $20, 21, 22, 23, 24, \dots, 2n, \dots$

Восьмеричная система: $80, 81, 82, 83, 84, \dots, 8n, \dots$

Пример. Десятичное число 4718,63, двоичное число 1001,1, восьмеричное число 7764,1, шестнадцатеричное число 3AF. Позиция цифры в числе называется разрядом: разряд возрастает справа налево, от младших к старшим, начиная с нуля.

Развернутая форма представления числа

В позиционной системе счисления любое вещественное число в развернутой форме может быть представлено в следующем виде:

$A = \pm (a_{n-1}q^{n-1} + a_{n-2}q^{n-2} + \dots + a_0q^0 + a_{-1}q^{-1} + a_{-2}q^{-2} + \dots + a_{-m}q^{-m})$ Здесь: A - само число, q - основание системы счисления, a_i - цифры, принадлежащие алфавиту данной системы счисления, n - число целых разрядов числа, m - число дробных разрядов числа. Развернутая форма записи числа - сумма произведений коэффициентов на степени основания системы счисления. Пример.

Десятичное число $A_{10} = 4718,63$ в развернутой форме запишется так:
 $A_{10} = 4 \cdot 10^3 + 7 \cdot 10^2 + 1 \cdot 10^1 + 8 \cdot 10^0 + 6 \cdot 10^{-1} + 3 \cdot 10^{-2}$

Двоичное число $A_2 = 1001,1 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1}$

Восьмеричное число $A_8 = 7764,1 = 7 \cdot 8^3 + 7 \cdot 8^2 + 6 \cdot 8^1 + 4 \cdot 8^0 + 1 \cdot 8^{-1}$

Шестнадцатеричное число $A_{16} = 3AF = 3 \cdot 16^2 + 10 \cdot 16^1 + 15 \cdot 16^0$

Системы счисления, используемые в вычислительной технике. Несмотря на то, что исторически человек привык работать в десятичной системе счисления, с технической точки зрения она крайне неудобна, так как в электрических цепях компьютера требовалось бы иметь одновременно десять различных сигналов. Тем не менее, такие схемы существуют в некоторых видах микрокалькуляторов. Чем меньше различных сигналов в электрических цепях, тем проще микросхемы, являющиеся основой конструкции большинства узлов ЭВМ, и тем надежнее они работают. Наименьшее основание, которое может быть у позиционных систем счисления это – двойка. Именно поэтому двоичная система счисления используется в вычислительной технике, а двоичные наборы приняты за средство кодирования информации.

В компьютере имеются только два устойчивых состояния работы микросхем, связанных с прохождением электрического тока через данное устройство (1) или его отсутствием (0). Говоря точнее, (1) кодирует высокое напряжение в схеме компьютера, а (0) – низкое напряжение. Если вспомнить, что двоичная система счисления обладает самыми маленькими размерами таблиц сложения и умножения, то можно догадаться, что этот факт должен сильно радовать конструкторов ЭВМ, поскольку обработка сигнала в этом случае будет также самой простой. Таким образом, двоичная система счисления, с точки зрения организации работы ЭВМ, является наилучшей. Мы уже говорили о преимуществах двоичной системы счисления с технической точки зрения организации работы компьютера.

Зачем нужны другие системы счисления, кроме, естественно, еще и десятичной, в которой человек привык работать? Чтобы ответить на него, возьмем любое число в десятичной системе счисления, например 255, и переведем его в другие системы счисления с основаниями, кратными двойке:

$$255_{10} = 11111111_2 = 3333_4 = 377_8 = FF_{16}.$$

Чем меньше основание системы счисления, тем больше разрядов требуется для его записи то есть, тем самым мы проигрываем в компактности записи чисел и их наглядности. Поэтому, наряду с двоичной и десятичной системами счисления, в вычислительной технике применяют так же запись чисел в 8-и 16-ричных системах счисления. Поскольку их основания кратны двойке, они органично связаны с двоичной системой счисления и преобразуются в эту систему наиболее быстро и просто (по сути они являются компактными видами записи двоичных чисел). Все другие системы счисления представляют для вычислительной техники чисто теоретический интерес.

Решение задач 1. Какое число записано с помощью римских цифр: CLVI

Решение: Зная обозначения, запишем: C – 100; L – 50; V – 5; I – 1

Пользуемся правилом записи чисел в римской системе счисления: Т.к. большая стоит перед меньшей – CL, то они складываются ($C+L = 100 + 50 = 150$).

Т.к. большая цифра стоит перед меньшей – VI, то они складываются ($V + I = 5 + 1 = 6$).

Следовательно, $150 + 6 = 156$ Ответ: CLVI = 15610

2. Записать в развернутом виде число: 3BFA16

Решение:

Пользуемся формулой: $A_q = \dots + a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + a_{-2} \cdot q^{-2} + \dots + a_{-m} \cdot q^{-m}$

В нашем случае: $a_1 = 3$; $a_2 = B$; $a_3 = F$; $a_4 = A$ $q=16$ $n=3, 2, 1, 0$

Следовательно: $3BFA_{16} = 3 \cdot 16^3 + B \cdot 16^2 + F \cdot 16^1 + A \cdot 16^0$

Ответ: $3BFA_{16} = 3 \cdot 16^3 + B \cdot 16^2 + F \cdot 16^1 + A \cdot 16^0$

3. Запишите в свернутой форме число $1 \cdot 8^2 + 4 \cdot 8^1 + 7 \cdot 8^0$

Решение: Пользуемся формулой: $A_q = \dots + a_{n-1} \cdot q^{n-1} + a_{n-2} \cdot q^{n-2} + \dots + a_0 \cdot q^0 + a_{-1} \cdot q^{-1} + a_{-2} \cdot q^{-2} + \dots + a_{-m} \cdot q^{-m}$

В нашем случае: $a_1 = 1$; $a_2 = 4$; $a_3 = 7$ $q=8$ $n= 2, 1, 0$

Следовательно: $1 \cdot 8^2 + 4 \cdot 8^1 + 7 \cdot 8^0 = 1478$

Ответ: $1 \cdot 8^2 + 4 \cdot 8^1 + 7 \cdot 8^0 = 1478$

4. Используя приложение Калькулятор операционной системы Windows запишите значения числа 1010₁₀ в различных системах счисления. Для этого: откройте калькулятор:

ПУСК-ПРОГРАММЫ-СТАНДАРТНЫЕ-КАЛЬКУЛЯТОР настройте вид калькулятора на инженерный: ВИД-ИНЖЕНЕРНЫЙ

Dec – десятичная система счисления

Oct – восьмеричная система счисления

Bin – двоичная система счисления

Hex – шестнадцатеричная система счисления поставьте флажок в Dec и наберите число 1010 поставьте флажок в Oct – вы увидите данное число, представленное в 8-ой системе счисления (запишите результат) поставьте флажок в Bin – вы увидите данное число, представленное в 2-ой системе счисления (запишите результат) поставьте флажок в Hex – вы увидите данное число, представленное в 16-ой системе счисления (запишите результат)

Алгоритмы перевода в системы счисления по разным основаниям

Алгоритм перевода чисел из любой системы счисления в десятичную Представить число в развернутой форме. При этом основание системы счисления должно быть представлено в десятичной системе счисления. Найти сумму ряда. Полученное число является значением числа в десятичной системе счисления. Алгоритм перевода целых чисел из десятичной системы счисления в любую другую Последовательно выполнять деление данного числа и получаемых целых частных на основание новой системы счисления

до тех пор, пока не получится частное, меньше делителя. Полученные остатки, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления. Составить число в новой системе счисления, записывая его, начиная с последнего остатка. Алгоритм перевода правильных дробей из десятичной системы счисления в любую другую.

Последовательно умножаем данное число и получаемые дробные части произведения на основание новой системы счисления до тех пор, пока дробная часть произведения не станет равна нулю или будет достигнута требуемая точность представления числа. Полученные целые части произведений, являющиеся цифрами числа в новой системе счисления, привести в соответствие с алфавитом новой системы счисления. Составить дробную часть числа в новой системе счисления, начиная с целой части первого произведения.

Алгоритм перевода произвольных чисел из десятичной системы счисления в любую другую. Перевод произвольных чисел, т.е. содержащих целую и дробную часть, осуществляется в два этапа: Отдельно переводится целая часть. Отдельно переводится дробная. В итоговой записи полученного числа целая часть отделяется от дробной запятой.

Тема 2.3. Поиск и сортировка

Сортировка массива — это расстановка элементов массива в некотором порядке. В отсортированном массиве, за счет предварительно выполненной работы по упорядочению, поиск элемента (даже в худшем случае) можно осуществлять, не просматривая весь массив. Пример отсортированного массива — это, например, список телефонов в справочнике, список фамилий в адресной книге и т. д.

Примером задачи сортировки может служить следующая: в массиве из N различных чисел необходимо осуществить их перестановку так, чтобы после нее в массиве первым элементом было самое большое число. Каждое следующее должно быть меньше предыдущего, а последнее — самое маленькое из чисел данного массива. Такой порядок расположения чисел называют **убывающим**.

Если в массиве есть равные числа, то задачу можно переформулировать следующим образом: в массиве из N чисел осуществить их перестановку так, чтобы после перестановки в массиве первым элементом было самое большое число. Каждое следующее должно быть не больше предыдущего, а последнее — самое маленькое из чисел данного массива. Такой порядок расположения чисел называют **невозрастающим**.

Порядок, при котором в массиве первым элементом является самое маленькое число, каждое следующее число больше предыдущего, а последнее — самое большое из чисел данного массива называют **возрастающим**.

Если в массиве есть равные числа и они расположены так, что первым элементом является самое маленькое число, каждое следующее число не меньше предыдущего, а последнее — самое большое из чисел данного массива, то такой порядок называют *неубывающим*.

Задача сортировки, как и любая другая задача, может решаться множеством способов, каждый из которых имеет как достоинства, так и недостатки. Выбор способа сортировки определяется особенностями решаемой задачи.

При выборе метода сортировки необходимо учитывать объем требуемой памяти и скорость работы. При сортировке массива желательно использовать как можно меньше дополнительной памяти, поэтому обычно рассматриваются алгоритмы, которые упорядочивают массив перестановками его элементов (без использования еще одного массива). Оценить скорость работы метода сортировки можно, оценив количество требуемых операций сравнения и (или) операций перестановок элементов.

Ниже рассматриваются методы сортировки линейного массива по убыванию (невозрастанию). Сортировка по возрастанию (неубыванию) производится аналогично.

Сортировка выбором

Давайте представим, что перед нами поставлена задача расставить N чисел по убыванию. Как бы мы ее решали?

Наверное, пришлось бы сначала найти максимальное из всех чисел и поменять местами с первым числом. Затем из еще неотсортированных элементов надо было бы опять выбрать максимальный элемент и поменять местами с первым элементом из еще не отсортированной части.

Примененный нами метод и называется *сортировкой выбором*.

Формально его можно описать следующим образом.

На i шаге ($1, \dots, N-1$): выбираем из элементов с индексами от i до N максимальный элемент; меняем местами найденный максимальный и элемент $A[i]$; на i -м месте оказывается максимальный элемент из еще неотсортированной части массива.

После выполнения $N-1$ -го шага в позиции $A[N]$ будет находиться самый маленький элемент массива.

Название метода, «*сортировка выбором*» определяется тем, что на каждом шаге мы находим (выбираем) максимальный элемент из еще неотсортированной части массива.

Запишем алгоритм сортировки выбором:

```

нц для i от 1 до N-1
| K := i
| max := A[i]
| нц для j от i+1 до N
| | если A[j] > max
| | | то
| | | max := A[j]
| | | K := j
| | все
| кц
A[K] := A[i]
A[i] := max
кц

```

Внутренний цикл ДЛЯ ОТ $i+1$ ДО N является ничем иным, как алгоритмом поиска максимального элемента среди элементов с номерами от $i+1$ ДО N .

Рассмотрим работу данного метода на массиве $N = \{0, 1, 9, 2, 4, 3, 6, 5\}$. Максимальный элемент будем подчеркивать.

0,	1,	9,	2,	4,	3,	6,	5
9,	1,	0,	2,	4,	3,	6,	5
9,	6,	0,	2,	4,	3,	1,	5
9,	6,	5,	2,	4,	3,	1,	0
9,	6,	5,	4,	2,	3,	1,	0
9,	6,	5,	4,	3,	2,	1,	0
9,	6,	5,	4,	3,	2,	1,	0
9,	6,	5,	4,	3,	2,	1,	0

Подсчитаем количество сравнений, которые пришлось сделать для упорядочения массива.

На первом шаге для нахождения максимального элемента необходимо $(N-1)$ сравнение, на втором $(N-2)$, на третьем $(N-3)$, ..., на последнем шаге — одно сравнение. Найдем сумму:

$$N-1 + N-2 + N-3 + \dots + 1 = N(N-1)/2 = (N^2 - N)/2.$$

Примечание. Сумму можно посчитать исходя из следующих соображений. Количество слагаемых равно $(N-1)$, сумма первого и последнего, второго и предпоследнего и т. д. равна. Произведение $(N-1)$ даст удвоенную сумму, так как каждое слагаемое будет входить в эту сумму дважды, поэтому его нужно разделить на 2,

Количество перестановок элементов равно $(N-1)!$. Это количество определяется внешним циклом ДЛЯ.

Сортировка обменом

Рассмотрим еще один метод сортировки, который формально можно описать так:

На i -м шаге ($i = 1, \dots, N-1$) выполняем:

Сравниваем первые два элемента. Если первый меньше второго, то меняем их местами.

Сравниваем второй и третий, третий и четвертый, ..., $N-1$ и $N-1+1$, при необходимости меняя элементы местами. Самый маленький окажется на i -м месте в массиве.

После первого шага самый маленький элемент массива помещается на N -е место. Массив будет отсортирован после просмотра, в котором участвуют только первый и второй элементы.

Название метода «*сортировка обменом*» определяется тем, что алгоритм основывается на обмене местами двух элементов массива.

Описанный метод сортировок обменом называют также *пузырьковой сортировкой*.

Алгоритм метода сортировки обменом:

```
нц для i от 1 до N-1
  нц для j от 1 до N-i
    если A[j] < A[j+1]
      то
        x := A[j]
        A[j] := A[j+1]
        A[j+1] := x
    все
  кц
кц
```

Число сравнений в данном алгоритме равно также $(N^2 - N)/2$.

При каждом сравнении возможна перестановка двух элементов в массиве. Поэтому количество перестановок (в худшем случае) будет равно количеству сравнений, т. е. $(N^2 - N)/2$.

На последних двух проходах в приведенном выше примере массив не менялся. Заметим, что если на каком-то шаге алгоритма элементы массива уже упорядочены, то при последующих проходах по массиву перестановки больше выполняться не будут. Следовательно, как только количество выполненных на последнем проходе перестановок станет равным 0, алгоритм можно заканчивать.

Если запоминать положение (индекс) K .

```

P := «Истина»
K := N - 1
нц пока P = «Истина»
| P := «Ложь»
| R := K
| нц для j от 1 до R
| | если A[j] < A[j + 1]
| | | то
| | | x := A[j]
| | | A[j] := A[j + 1]
| | | A[j + 1] := x
| | | P := «Истина»
| | | K := j
| | все
| кц
кц

```

В приведенном фрагменте переменная P логического типа используется для определения, были перестановки или нет, а переменная K — для хранения индекса последнего обмена. Переменная J является границей, на которой заканчивается просмотр.

Если проанализировать пузырьковую сортировку, то можно заметить, что самое маленькое число занимает свое место за один проход по массиву, а самое большое перемещается по направлению к своему месту на одну позицию при каждом проходе. Это наводит на мысль чередовать направление проходов. Такая сортировка называется *шейкер-сортировкой*. Рассмотрим ее работу на том же массиве $N = \{0, 1, 9, 2, 4, 3, 6, 5\}$. (R — левая граница просмотра, L — правая.)

1,	9,	2,	4,	3,	6,	5,	0
9,	1,	6,	2,	4,	3,	5,	0
9,	6,	2,	4,	3,	5,	1,	0
9,	6,	5,	2,	4,	3,	1,	0
9,	6,	5,	4,	3,	2,	1,	0
9,	6,	5,	4,	3,	2,	1,	0

Алгоритм данной сортировки попытайтесь написать самостоятельно.

Все эти улучшения сокращают количество операций сравнения для частных случаев, однако при неблагоприятной начальной расстановке элементов массива (подумайте какой) приходится проделать все $(N^2 - N)/2$ операции сравнения.

Вопросы для повторения

1. На каких принципах основана сортировка выбором?
2. На каких принципах основана сортировка обменом?

3. За какое количество операций сравнения будет отсортирован массив, если применять сортировку обменом? сортировку выбором?
4. Сколько перестановок будет сделано для упорядочения массива, если применять сортировку обменом? сортировку выбором?
5. Как можно сократить время работы алгоритма сортировки обменом?
6. Алгоритм, какой из сортировок будет работать быстрее? Почему?

Тема 2.4. Задачи комбинаторики. Графы

При решении практических задач часто приходится выбирать из некоторого конечного множества объектов подмножество элементов, обладающих теми или иными свойствами, размещать элементы множеств в определенном порядке и т. д. Эти задачи принято называть комбинаторными задачами.

Многие из комбинаторных задач решаются с помощью двух основных правил – суммы и произведения. Если удастся разбить множество объектов на классы и каждый объект входит в один и только один класс, то общее число объектов равно сумме числа объектов во всех классах (правило сложения).

Правило умножения. Даны два произвольных конечных множества объектов A и B . Количество объектов в A равно N , в B – M . Составляются упорядоченные пары ab , где $a \in A$ (a принадлежит A) и $b \in B$. Количество таких пар (объектов) равно $N \times M$. Правило обобщается и на большее количество множеств объектов.

Перестановки (P_N – число перестановок). Классической задачей комбинаторики является задача о числе перестановок – сколькими способами можно переставить N различных предметов, расположенных на N различных местах.

N различных предметов, расположенных на N различных местах, можно переставить $N!$ (N -факториал) $= 1 \times 2 \times 3 \times \dots \times N$ способами ($P_N = N!$).

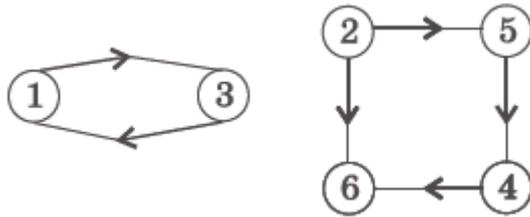
Отвлекаясь от природы объектов, можно считать их для простоты натуральными числами от 1 до N . При этой трактовке понятие перестановки можно дать следующим образом. Перестановкой из N элементов называется упорядоченный набор из N различных чисел от 1 до N . Места также нумеруются от 1 до N . Тогда перестановку можно записать таблицей с двумя строками, из которых каждая содержит все числа от 1 до N .

Например,

1 2 3 4 5 6

3 5 1 6 4 2

Каждую перестановку можно представить графически в виде ориентированного графа с множеством вершин от 1 до N .



Размещения (A_N^M) . Задача формулируется следующим образом: сколькими способами можно выбрать и разместить по M различным местам M из N различных предметов?

Число размещений вычисляется по формуле:

$$A_N^M = N \times (N - 1) \times (N - 2) \times \dots \times (N - M + 1),$$

$$\text{или } A_N^M = \frac{N!}{(N - M)!}$$

Сочетания (выборки) (C_N^M) . Другой классической задачей комбинаторики является задача о числе выборок: сколькими способами можно выбрать M из N различных предметов?

Сочетание можно определить и как подмножество из M различных натуральных чисел, принадлежащих множеству чисел из интервала от 1 до N . Число сочетаний вычисляется по формуле:

$$C_N^M = \frac{N!}{M! \times (N - M)!}$$

Полезные формулы:

$$C_N^0 = C_N^N = 1 \quad (N \geq 1)$$

$$C_N^M = C_N^{N-M} \quad (\text{правило симметрии})$$

$$C_N^M = C_{N-1}^{M-1} + C_{N-1}^M \quad (N > 1, 0 < M < N) \quad (\text{правило Паскаля})$$

$$C_N^M \times C_M^K = C_N^K \times C_{N-K}^{M-K}$$

$\sum_{i=0}^N C_N^i = 2^N$. Эта формула является следствием известного соотношения под названием бином Ньютона:

$$(x + y)^N = \sum_{k=0}^N C_N^k x^k y^{N-k}.$$

При $x=y=1$ бином Ньютона дает приведенную формулу.

Размещения с повторениями (A_N^K) .

Даны предметы, относящиеся к N различным типам. Из них составляются всевозможные расстановки длины k . При этом в расстановки могут входить и предметы одного типа. Две расстановки считаются различными, если они отличаются или типом входящих в них предметов, или порядком этих предметов.

Общая формула для подсчета числа размещений с повторениями:

$$A_N^k = N^k$$

Перестановки с повторениями. Выше рассмотрены перестановки из различных предметов. В том случае, когда некоторые переставляемые предметы одинаковы, часть перестановок совпадает друг с другом. Перестановок становится меньше. Задача формулируется следующим образом. Имеются предметы k различных типов. Сколько перестановок можно сделать из N_1 предметов первого типа, N_2 – второго, ..., N_k – k -го типа ($N_1+N_2+\dots+N_k=N$)?

Общая формула для подсчета перестановок с повторениями имеет вид:

$$P(N_1, N_2, \dots, N_k) = \frac{N!}{N_1! \times N_2! \times \dots \times N_k!},$$

где $N = N_1 + N_2 + \dots + N_k$ – общее число элементов.

Сочетания с повторениями. Имеются предметы N различных типов. Сколько существует расстановок длины k , если не различать порядок предметов в расстановке (различные расстановки должны отличаться хотя бы одним предметом)?

Общая формула подсчета числа сочетаний с повторениями имеет вид:

$$\bar{C}_N^k = C_{N+k-1}^{N-1} = C_{N+k-1}^k$$

Рассмотрим программную реализацию.

Размещения с повторениями:

```
for i:=1 to n do
  for j:=1 to n do
    writeln (A[i], A[j]);
```

Размещения без повторений:

```
for i:=1 to n do
  for j:=1 to n do
    if i<>j then writeln (A[i], A[j]);
```

Сочетания с повторениями

```
for i:=1 to n do
  for j:=i to n do
    writeln (A[i], A[j]);
```

Сочетания без повторений:

```
for i:=1 to n-1 do
  for j:=i+1 to n do
    writeln (A[i], A[j]);
```

ПРАКТИЧЕСКИЙ РАЗДЕЛ

РАЗДЕЛ 2. МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ

Тема 2.1. Общая характеристика методов и способов решения задач по информатике

Лабораторная работа № 1.

1. Для нормального разведения золотых рыбок необходимо, чтобы на каждую рыбку в аквариуме приходилось не менее 3-х литров воды. По известным объему аквариума и количеству рыбок, в нем содержащихся, определить, является ли аквариум "перенаселенным" или нет, и указать количество рыбок, которых в случае перенаселенности необходимо поместить в другой аквариум.

2. По данным статистического исследования, производительность птицефермы такова, что каждые полторы курицы за полтора дня сносят полтора яйца. Написать программу, которая позволяет рассчитать, сколько яиц (без десятых долей) снесут N кур за d дней (N и d - целые числа).

3. В расписании движения поездов указано время отправления 12 пригородных поездов со станции г. Урюпинска. Определить количество поездов, отправляющихся со станции в период времени с 16.00 до 19.30. (Время отправления поездов задается одномерным массивом.)

4. Дано неравенство: $50010 > 1000N - 100N > 25010$, где N – основание системы счисления. Определите, для каких N верно данное неравенство. В ответе запишите через пробел все подходящие основания систем счисления в порядке возрастания.

5. В телевидении для уменьшения нагрузки на канал передачи данных существует два формата передачи кадра изображения: 1. Прогрессивная развертка – все точки кадра передаются сразу в рамках одного пакета. 2. Чересстрочная развертка – кадр разбивается на два полукадра, составленных из четных и нечетных строк кадра соответственно. Полукадры передаются по каналу передачи данных двумя последовательными пакетами. Определите высоту передаваемого устройству кадра в точках, если известно, что: 1. Ширина кадра 4096 точек. 2. Используется палитра в 232 цветов. 3. Разница между пакетами прогрессивной и чересстрочной разверток составляет ровно 13 712 КБайт. 4. Высота кадра составляет четное количество точек. В ответе укажите целое число точек. Примечание: 1 КБайт=1024 байта.

Итерационные методы

а) Решить систему $Ax = b$, методом Гаусса с постолбцовым выбором главного элемента.

б) Выполнить LU разложение данной матрицы и с его помощью получить решение данной системы, а также вычислить определитель матрицы A .

с) Вычислить число обусловленности $\text{cond}(A)$ и исследовать чувствительность данной системы к погрешностям исходных данных.

Варианты к 1, 2 заданиям

1.

$$\begin{cases} -12.73x_1 + 37.45x_2 - 26.05x_3 + 0.52x_4 = 0.09, \\ -65.26x_1 + 104.13x_2 + 11.69x_3 + 32.57x_4 = 0.16, \\ -5.18x_1 + 67.52x_2 - 53.30x_3 - 19.30x_4 = 0.33, \\ -31.82x_1 + 35.24x_2 + 15.50x_3 + 23.36x_4 = 2.23. \end{cases}$$

2.

$$\begin{cases} 0.85x_1 + 0.50x_2 + 1.41x_3 + 0.43x_4 = 1.05, \\ 1.15x_1 + 0.88x_2 + 0.66x_3 + 1.07x_4 = 0.10, \\ 1.43x_1 + 1.50x_2 + 0.46x_3 + 1.05x_4 = 0.03, \\ 0.13x_1 + 1.20x_2 + 1.45x_3 + 1.47x_4 = 1.38 \end{cases}$$

3.

$$\begin{cases} -3.22x_1 + 11.08x_2 - 16.29x_3 - 2.54x_4 = 1.08, \\ -22.63x_1 + 50.14x_2 + 0.15x_3 + 13.79x_4 = 2.27, \\ -10.41x_1 + 59.91x_2 - 30.05x_3 - 9.09x_4 = 0.37, \\ -36.64x_1 + 74.51x_2 - 18.71x_3 + 12.21x_4 = 0.24. \end{cases}$$

Задание 1. «Арифметический квадрат». Сколько путей ведет из клетки (1, 1) в клетку (N, M), с условием, что двигаться можно только вниз или вправо?

Задание 2. Выведите на экран «Треугольник Паскаля».

Задание 3. Раскрыть скобки в выражении: $(A+B)^n$ (n ввести с клавиатуры).

Задание 4. Найти кратчайший путь в двумерном массиве из клетки 1,1 в клетку N M. (путем называется набор индексов проходимых элементов, кратчайшим путем - набор индексов тех элементов массива, сумма значений которых минимальна). Двигаться можно только вниз либо вправо.

Задание 5. Дана матрица. Составьте программу ее заполнения в соответствии с приведенным шаблоном. Количество строки и столбцов вводится с клавиатуры.

Тема 2.2. Задачи обработки чисел

Лабораторная работа № 2.

а) Переведите число n , представленное в десятичной системе счисления, в систему счисления с основанием a .

вариант	n	a	вариант	n	a	вариант	n	a
1	1356	3	11	3971	5	21	7744	6
2	2426	5	12	11135	9	22	12221	6
3	5759	7	13	9971	7	23	2173	9
4	1097	4	14	10992	8	24	8765	3
5	12345	9	15	5159	4	25	7432	7
6	6876	6	16	9786	6	26	7356	6
7	9112	8	17	2234	3	27	2083	5
8	8971	7	18	13133	9	28	4999	3
9	2099	3	19	2596	7	29	5001	4
10	4396	4	20	9432	4	30	4492	8

б) Переведите число m из системы счисления с основанием b в десятичную систему счисления.

вариант	m	b	вариант	m	b	вариант	m	b
1	7814	9	11	31330	4	21	7145	9
2	210122	3	12	40432	5	22	130122	4
3	7715	8	13	221110	3	23	220021	3
4	103122	4	14	6805	9	24	42031	5
5	6354	7	15	7155	8	25	5543	6
6	122021	3	16	12340	5	26	6776	8
7	25441	6	17	5654	7	27	6321	7
8	8135	9	18	30032	4	28	33021	4
9	33421	5	19	11222	3	29	30444	5
10	6503	7	20	6733	8	30	3871	9

Вычислите сумму и разность двух чисел

а) в двоичной системе счисления:

вариант			вариант			вариант		
1	101101	10011	11	101101	11010	21	100001	10111
2	110101	11100	12	110110	10011	22	111010	11001
3	111011	10011	13	100101	11110	23	101110	11000
4	101011	11100	14	111010	11010	24	100101	10111
5	100111	10110	15	111011	10001	25	110101	11110
6	101101	11001	16	100001	11001	26	101011	11001
7	111100	10110	17	110110	11110	27	101010	10001
8	111001	11001	18	111110	10011	28	110011	11100
9	110010	10111	19	101101	11001	29	110001	11010
10	101011	11001	20	100011	11010	30	101110	10110

б) в восьмеричной системе счисления:

вариант			вариант			вариант		
1	7315	5142	11	5177	4114	21	5427	4177
2	7221	4473	12	7017	3333	22	4316	2612
3	6513	4546	13	7314	4766	23	6423	5074
4	7176	3467	14	7532	6516	24	6131	3777
5	6162	5244	15	5417	3427	25	7076	5146
6	6427	3175	16	7362	1563	26	6113	4412
7	7071	2254	17	6422	2714	27	6117	5534
8	5615	3716	18	7174	6143	28	4213	3763
9	7365	4142	19	5345	4715	29	5411	4154
10	6423	5561	20	4721	3223	30	6666	2117

Задачи целочисленной арифметики

1. Дано натуральное число n . Выбросить из записи числа все четные цифры.
2. Дано натуральное число n . Определить, является ли оно палиндромом (читается одинаково с начала и с конца).
3. Найти все числа из диапазона от n до m , которые при возведении в квадрат дают палиндром.
4. Найти все числа-палиндромы из диапазона от n до m , которые при возведении в квадрат также дают палиндромы.
5. Найти остаток от деления числа, записываемого с помощью k семерок, на число a , k и a — заданные натуральные числа.
6. Все натуральные числа выписаны подряд, начиная с единицы. Определить, какая цифра стоит на N -м месте.
7. На интервале $(1000; 9999)$ найти все простые числа, каждое из которых обладает тем свойством, что сумма первой и второй цифр записи этого числа равна сумме третьей и четвертой.
8. Высадившись на планете Альфа, населенной разумными «осьминогами», космонавты увидели написанную на стене близлежащего строения формулу $99 \cdot 99 = 1210$. Сколько щупальцев было у населявших планету Альфа «осьминогов»?

Тема 2.3. Поиск и сортировка

Лабораторная работа № 3.

Задание 1. Отсортировать строки массива целых чисел по убыванию. Сортировка включением.

Задание 2. Отсортировать элементы квадратной вещественной матрицы размерности n , стоящие на главной диагонали, применив сортировку бинарным включением.

Задание 3. Отсортировать положительные элементы одномерного массива, отрицательные оставить на местах. Пузырьковая сортировка.

Задание 4. Заданный одномерный массив отсортировать по возрастанию цифры десятков каждого элемента. Сортировка прямой выбор.

Задание 5. Упорядочить одномерный массив так, чтобы в начале располагались четные элементы в порядке возрастания их значений, а затем нечетные — в порядке убывания их значений.

Тема 2.4. Задачи комбинаторики. Графы

Лабораторная работа № 4.

1. Сколькими способами можно раскрасить вершины куба в три цвета (например, красный, синий и зеленый)? Напечатать все возможные способы.

2. Сколькими различными способами можно надеть на нить семь бусин двух цветов — синего и белого? Напечатать все возможные варианты.

3. Сколько различных ожерелий из семи бусин можно составить из бусин двух цветов — синего и белого? (Под ожерельем понимается замкнутая нить с нанизанными бусинами.) Напечатать все возможные варианты.

4. Сколько различных ожерелий можно составить из двух белых, двух синих и двух красных бусин? Напечатать все возможные варианты.

5. Сколькими различными способами можно грани куба раскрасить в четыре цвета? Напечатать все возможные варианты.

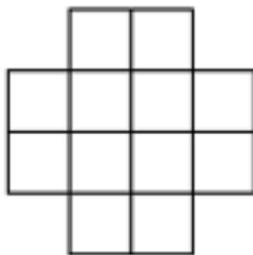
6. Грани куба можно раскрасить: а) все в белый цвет; б) все в черный цвет; в) часть в белый, а остальные в черный. Сколько имеется различных способов раскраски? Напечатать все возможные варианты.

7. Напечатать все четырехзначные десятичные числа, у которых все цифры разные. Обобщить на N -значные числа.

Использование видов соединений при решении задач повышенной сложности по информатике.

1. Между девятью планетами солнечной системы установлено космическое сообщение. Рейсовые ракеты летают по следующим маршрутам: Земля – Меркурий; Плутон – Венера; Земля – Плутон; Плутон – Меркурий; Меркурий – Вене; Уран – Нептун; Нептун – Сатурн; Сатурн – Юпитер; Юпитер – Марс и Марс – Уран. Можно ли долететь на рейсовых ракетах с Земли до Марса?

2. Доска имеет форму двойного креста, который получается, если из квадрата 4×4 убрать угловые клетки.

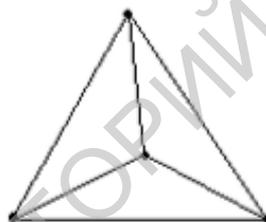


Можно ли обойти ее ходом шахматного коня и вернуться на исходную клетку, побывав на всех клетках ровно по одному разу?

3. В городе Маленьком 15 телефонов. Можно ли их соединить проводами так, чтобы каждый телефон был соединен ровно с пятью другими?

4. В Тридевятом царстве только один вид транспорта – ковер-самолет. Из столицы выходит 21 ковровиния, из города Дальний – одна, а из всех остальных городов, – по 20. Докажите, что из столицы можно долететь в город Дальний.

5. Можно ли нарисовать изображенный на рисунке граф не отрывая карандаш от бумаги и проводя каждое ребро ровно один раз?



РАЗДЕЛ 3. РЕШЕНИЕ ПРАКТИКООРИЕНТИРОВАННЫХ ЗАДАЧ. ТВОРЧЕСКИЙ ПРОЕКТ: ПОСТАНОВКА ЗАДАЧИ, ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

Тема 3.1. Творческий проект: постановка практикоориентированной задачи

Лабораторная работа № 5.

Цель: продумать постановку задачи, представленной на естественном языке с учетом конкретной предметной области, провести системный анализ задачи.

Задание 1. Предлагается самостоятельно выбрать темы творческих проектов из любой предметной области информатики.

Задание 2. Предлагается самостоятельно выбрать темы творческих проектов из любой предметной области математики.

Задание 3. Предлагается самостоятельно выбрать темы творческих проектов из любой предметной области физики.

Задание 4. Предлагается самостоятельно выбрать темы творческих проектов из любой предметной области биологии.

Задание 5. Предлагается самостоятельно выбрать темы творческих проектов из любой предметной области химии.

Лабораторная работа № 6.

Цель: рассмотреть примеры геометрических задач, в основе решения которых лежит вычисление площади треугольника, научиться применять изученные методы при решении геометрических задач.

Задание 1. Найти площадь треугольника, заданного координатами своих вершин.

Задание 2. Определить площадь выпуклой фигуры, заданной координатами своих вершин.

Задание 3. Определить - находится точка внутри или вне выпуклого многоугольника, заданного координатами своих вершин.

Задание 4. Определить, пересекаются ли два отрезка, заданные на плоскости координатами своих концов.

Задание 5. С клавиатуры вводятся пары координат n точек на плоскости. Найти длину ломанной, соединяющей эти точки. Начало ломанной - ближайшая к центру координатной плоскости точка, ломанная соединяет все точки кратчайшим способом.

Тема 3.2. Использование современных технологий и средств программирования для обработки и визуализации данных

Лабораторная работа № 7.

Цель: рассмотреть визуализацию моделей в популярной системе математических расчетов при решении задач из разных предметных областей.

Задание 1. В MathCad решить систему уравнений:
$$\begin{cases} y^2 + x^2 - 7 = 0 \\ y - x = 2. \end{cases}$$
 Дать графическую иллюстрацию. Показать на графике точки пересечения кривых, задаваемых уравнениями.

Задание 2. В MathCad составить уравнение окружности радиуса 2 с центром в точке локального минимума функции $f(x) = x^3 - 25x^2 - x + 2.7$. Построить график кривой y у показать окружность и ее центр.

Задание 3. В MathCad составить уравнение окружности с центром в точке $(2, 0)$ и касающейся прямой $y = 3x$. Найти координаты точки касания. Дать графическую иллюстрацию, показать точку касания и центр окружности.

Задание 4. В MathCad для функции $Y = x^3 - 3x$ на отрезке $[-3; 3]$ найти точки локальных экстремумов и нули функции. Дать графическую иллюстрацию, показать точки экстремумов.

Задание 5. В MathCad составить уравнение окружности, проходящей через точки (3, 2), (7, 4), если ее центр лежит на прямой $3x - 5y + 7$. Дать графическую иллюстрацию, показать центр окружности.

Построить график функции $y = \sin(x)+2$ на отрезке $[-6; 6]$ с шагом $h =$

1. Объяснить изменения графика
2. Построить график функции $y=4x+4x^2-x^3$ на интервале от $[-4;4]$ с шагом -1.

Задана произвольная элементарная функция аналитическим выражением.

Протабулировать функцию для значений аргумента, изменяющегося на отрезке $[a, b]$ с шагом $h=(a - b)/n$, т.е. разбив отрезок на n частей и построить график функции, используя

- а) интегрированную среду программирования TPascal;
- б) электронную таблицу Excel;
- в) программу MathCad

Лабораторная работа № 8.

Цель: систематизировать знания по использованию Mathcad, выполнить разноуровневые индивидуальные задания.

Задание 1. Создать массив из 200 случайных чисел на отрезке $[1; 10]$. Найти минимальное, максимальное и среднее значения. Выполнить выборку чисел, входящих в отрезок $[5,4; 7,56]$, вычислить их сумму и количество. Построить столбиковую диаграмму распределения случайных чисел исходного массива по интервалам $(1 ;2)$, $(2;3)$ и т. д.

Задание 2. Построить график траектории движения тела брошенного под углом к горизонту, если начальная скорость $=500$ м/сек, а угол броска $\theta=55^\circ$. Определить угол броска, при котором дальность движения по горизонтали будет максимальной. Указание: задачу сформулировать в общем виде.

Лабораторная работа № 9.

Цель: систематизировать знания по использованию Mathcad, выполнить разноуровневые индивидуальные задания.

Задание 1. Создать массив из 200 случайных чисел на отрезке $[1; 10]$. Найти минимальное, максимальное и среднее значения. Выполнить выборку чисел, входящих в отрезок $[5,4; 7,56]$, вычислить их сумму и количество. Построить столбиковую диаграмму распределения случайных чисел исходного массива по интервалам $(1 ;2)$, $(2;3)$ и т. д.

Задание 2. Построить график траектории движения тела брошенного под углом к горизонту, если начальная скорость $=500$ м/сек, а угол броска $\theta=55^\circ$. Определить угол броска, при котором дальность движения по горизонтали будет максимальной. Указание: задачу сформулировать в общем виде.

Тема 3.3. Творческий проект: верификация данных, защита

Лабораторная работа № 10.

Цель: верификация результатов работы над творческим проектом.

Задание. Тестирование проекта. Оценка эффективности и результативности работы программного средства. Защита проекта.

РЕПОЗИТОРИЙ БГПУ

РАЗДЕЛ КОНТРОЛЯ ЗНАНИЙ

Вопросы к зачету

1. Этапы решения задач с помощью компьютера.
2. Классификация задач по информатике.
3. Анализ проблем решения задач в теории и практике обучения информатике.
4. Процесс решения задач по информатике.
5. Обзор задач, предлагаемых в школьном курсе информатики.
6. Особенности становления обобщенного умения по решению задач из курса информатики.
7. Олимпиадные задачи по информатике. Способы решения олимпиадных задач по информатике.
8. Понятие задачи и подзадачи. Сведение задачи к подзадачам.
9. Итерационные методы решения задач.
10. Рекурсия и рекуррентные соотношения.
11. Методы динамического программирования при решении олимпиадных задач по информатике.
12. Задачи обработки чисел.
13. Позиционные и непозиционные системы счисления. Перевод чисел из одной позиционной системы счисления в другую.
14. Задачи целочисленной арифметики. Выделение цифр числа двумя способами.
15. Операции с дробями, сокращение дробей.
16. Действия с многозначными числами: сложение, вычитание и произведение многозначных чисел.
17. Цепные дроби. Разложение рациональных и иррациональных чисел в цепные дроби.
18. Поиск. Алгоритмы поиска.
19. Сортировка. Алгоритмы сортировки.
20. Задачи комбинаторики.
21. Комбинаторика, понятие об упорядочении и группировании объектов.
22. Перестановки. Алгоритм подсчета и генерации перестановок.
23. Сочетания. Алгоритм подсчета и генерации сочетаний.
24. Размещения. Алгоритм подсчета и генерации размещений.

ВСПОМОГАТЕЛЬНЫЙ РАЗДЕЛ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Качество профессионально-методической подготовки будущего учителя информатики в значительной степени определяется уровнем профессиональной компетентности в сфере информатики. Одним из параметров, который определяет это является уровень форсированности обобщенных умений решать задачи из цикла специальных дисциплин специальностей «Математика и информатика», «Физика и информатика».

Систематизация содержания основных направлений информатики как фундаментальной науки, так и интенсивно развивающегося школьного учебного предмета – одна из главных задач учебной дисциплины **«Практикум по решению задач по информатике»**. Возможности и опыт, приобретаемый студентами в процессе изучения данной учебной дисциплины, позволят им осуществлять междисциплинарные практикоориентированные связи, поддерживать идеи профильного обучения в школе, развивать олимпиадное движение и исследовательскую деятельность по информатике, а также оптимизировать организационно-управленческий потенциал учебного процесса в системе образования.

Цель учебной дисциплины – формирование у будущих учителей информатики профессиональных компетенций для решения содержательно-деятельностных задач учебного процесса, обобщение знаний методов и современных подходов решения задач по информатике.

Основные задачи учебной дисциплины:

- формирование целостного системного представления о структуре процесса решения задач по информатике;
- обобщение знаний методов и способов решения задач по информатике;
- применение оптимальных методов и современных технологий к решению профессиональных задач;
- эффективная реализация творческого подхода при определении методов и средств решению задач информатики;
- мотивация к осуществлению самообразования и совершенствования в условиях интенсивного развития IT-сферы.

Место учебной дисциплины в системе подготовки специалиста

Изучение учебной дисциплины **«Практикум по решению задач по информатике»** опирается на основные академические, социально-личностные и профессиональные компетенции, сформированные у студентов в процессе изучения ими таких учебных дисциплин как «Педагогика», «Психология», «Технологии программирования и методы алгоритмизации», «Вычислительные методы и компьютерное моделирование», «Методика преподавания информатики». Благодаря ее изучению формируется целостное

представление о методах и современных подходах к решению задач по информатике.

Профессиональные компетенции студента

Учебная дисциплина «**Практикум по решению задач по информатике**» входит в компонент учреждения высшего образования. Изучение учебной дисциплины «Практикум по решению задач по информатике» должно способствовать формированию у студентов академических, социально-личностных и профессиональных компетенций.

Требования к академическим компетенциям

Специалист должен:

- АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.
- АК-Ч. Уметь работать самостоятельно.
- АК-5. Быть способным порождать новые идеи (обладать креативностью).
- АК-б. Владеть междисциплинарным подходом при решении проблем.
- АК-10. Уметь осуществлять учебно-исследовательскую деятельность.

Требование к социально-личностным компетенциям

Специалист должен:

- СЛК-4. Владеть навыками здоровьесбережения.
- СЛК-6. Уметь работать в команде.
- СЛК-7. Быть способным к осуществлению самообразования и самосовершенствования профессиональной деятельности.

Требования к профессиональным компетенциям

Специалист должен быть способен:

Обучающая деятельность

- ПК-1. Управлять учебно-познавательной и учебно-исследовательской деятельностью обучающихся.
- ПК-2. Использовать оптимальные методы, формы, средства обучения.
- ПК-Ч. Организовывать самостоятельную работу обучающихся.

Воспитательная деятельность

- ПК-В. Формировать базовые компоненты культуры личности обучающегося.

Развивающая деятельность

- ПК-121 (ПК-1 12). Развивать навыки самостоятельной работы обучающихся с учебной, справочной, научной литературой и др. источниками информации.

Ценностно-ориентационная деятельность

– ПК-17. Осуществлять профессиональное самообразование и самовоспитание с целью совершенствования профессиональной деятельности.

В результате изучения учебной дисциплины студент должен знать:

- структуру процесса решения задач с помощью компьютера;
- методы и способы решения задач по информатике.

В результате изучения учебной дисциплины студент должен уметь:

- проводить системный анализ задачи, представленной на естественном языке, с учетом конкретной области деятельности человека;
- выбирать оптимальные методы и средства решения задачи;
- реализовывать все этапы решения задачи на компьютере;
- осуществлять верификацию полученных результатов.

В результате изучения учебной дисциплины студент должен владеть:

- методами поиска, анализа и дидактической адаптации научной информации по решению задач по информатике;
- современными технологиями и средствами для решения профессиональных задач.

Структура и содержание учебной дисциплины Учебная дисциплина **«Практикум по решению задач по информатике»** изучается на протяжении одного семестра и содержит три раздела. В первом разделе осуществляется теоретическая классификация видов задач и целостное, системное представление о структуре процесса решения задач по информатике. Во втором разделе рассматриваются методы и способы решения задач по информатике. Третий раздел посвящен реализации творческого проекта, предполагающего использование современных технологий и средств программирования при решении практикоориентированных задач и визуализации данных.

Данная учебная программа является основным документом, определяющим объем и содержание учебной дисциплины **«Практикум по решению задач по информатике»** для специальности 1-02 05 02 Физика и информатика.

Методы обучения

Обучение учебной дисциплине проходит в рамках организации лекционных и лабораторных занятий. При чтении лекций особое внимание следует уделять использованию мультимедийных технологий.

Организация лабораторных занятий предполагает использование лично-ориентированных методов обучения, что способствует развитию индивидуально-творческих способностей студентов и приобретению умений самостоятельной работы. Лабораторные работы направлены на формирование навыков решения профессионально-методических задач учителя информатики.

Содержание и формы самостоятельной работы студентов разрабатываются в соответствии с целями и задачами подготовки специалиста. Среди видов самостоятельной работы студентов представляется возможным применять: самостоятельную работу во время основных аудиторных занятий (лекций, лабораторных занятий); самостоятельную работу в форме консультаций; внеаудиторную самостоятельную работу при выполнении студентами домашних заданий учебного и творческого характера. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

Распределение общего количества часов по формам обучения и семестрам

Специальность 1-02 05 02 Физика и информатика

Дневная форма получения высшего образования:

Всего на учебную дисциплину - 86 часов.

6 семестр — 42 часа аудиторных (8 часов — лекции, 34 часа - лабораторные занятия), 44 часа — самостоятельная работа.

Форма контроля — зачет (6 семестр).

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

РАЗДЕЛ I. ПРОЦЕСС РЕШЕНИЯ ЗАДАЧ С ПОМОЩЬЮ КОМПЬЮТЕРА

Тема 1.1. Этапы решения задач с помощью компьютера

Классификация задач по информатике. Анализ проблем решения задач в теории и практике обучения информатике. Процесс решения задач по информатике. Моделирование и формализация. Постановка задачи. Построение модели. Разработка алгоритма. Реализация алгоритма. Тестирование и отладка. Анализ результатов.

РАЗДЕЛ II. МЕТОДЫ И СПОСОБЫ РЕШЕНИЯ ЗАДАЧ ПО ИНФОРМАТИКЕ

Тема 2.1. Общая характеристика методов и способов решения задач по информатике

Обзор задач, предлагаемых в школьном курсе информатики. Особенности становления обобщенного умения по решению задач из курса информатики.

Олимпиадные задачи по информатике. Способы решения олимпиадных задач по информатике.

Понятие задачи и подзадачи. Сведение задачи к подзадачам.

Итерационные методы: решето Эратосфена, схема Горнера, числа Фибоначчи, числовые ряды и т.д.

Рекурсия и рекуррентные соотношения.

Динамическое программирование. Методы динамического программирования при решении олимпиадных задач по информатике.

Тема 2.2. Задачи обработки чисел

Позиционные и непозиционные системы счисления. Перевод чисел из одной позиционной системы счисления в другую. Кодирование различных видов информации.

Задачи целочисленной арифметики. Выделение цифр числа двумя способами. Операции с дробями, сокращение дробей.

Действия с многозначными числами: сложение, вычитание и произведение многозначных чисел.

Цепные дроби. Разложение рациональных и иррациональных чисел в цепные дроби.

Тема 2.3. Поиск и сортировка

Произвольный поиск в произвольно организованных структурах. Поиск в упорядоченно организованных структурах. Сокращение областей поиска.

Методы сортировки: простой вставкой, бинарной вставкой, слияниями (алгоритм фон Неймана) и др.

Тема 2.4. Задачи комбинаторики. Графы

Комбинаторика, понятие об упорядочении и группировании объектов. Виды соединений (комбинаций): перестановки, сочетания, размещения. Соединения с повторениями. Реализация перебора вариантов. Использование видов соединений при решении задач повышенной сложности по информатике.

Графы. Алгоритмы решения задач на графах.

РАЗДЕЛ III. РЕШЕНИЕ ПРАКТИКООРИЕНТИРОВАННЫХ ЗАДАЧ. ТВОРЧЕСКИЙ ПРОЕКТ: ПОСТАНОВКА ЗАДАЧИ, ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ

Тема 3.1. Творческий проект: постановка практико-ориентированной задачи

Постановка задачи, представленной на естественном языке, с учетом конкретной предметной области. Системный анализ задачи.

Тема 3.2. Использование современных технологий и средств программирования для обработки и визуализация данных

Геометрические задачи. Представление геометрических объектов на плоскости: точка, отрезок, прямая. Взаимное расположение геометрических объектов. Треугольники. Многоугольники. Площади фигур. Взаимное размещение фигур на плоскости.

Визуализация объектов в современных средах программирования при решении задач из разных предметных областей.

Построение графиков математических функций. Моделирование и визуализация физических законов, процессов и явлений.

Тема 3.3. Творческий проект: верификация результатов, защита

Тестирование проекта. Оценка эффективности и результативности работы программного средства. Защита проекта.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА
 для специальности 1-02 05 02 Физика и информатика
 для дневной формы получения образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов				Количество часов самостоятельная работа	Форма контроля знаний
		лекции	Лабораторные занятия	Практические занятия	Иное		
1	2	3	5	6	7		
1.	Процесс решения задач с помощью компьютера	2					
1.1	Этапы решения задач с помощью компьютера 1. Классификация задач по информатике. 2. Анализ проблем решения задач в теории и практике обучения информатике. 3. Процесс решения задач по информатике. Моделирование и формализация. Постановка задачи. Построение модели. Разработка алгоритма. 4. Реализация алгоритма. Тестирование и отладка. Анализ результатов.	2					Устный опрос, компьютерное тестирование
2.	Методы и способы решения задач по информатике	4	22			32	
2.1	Общая характеристика методов и способов решения задач по информатике	2	6			10	
2.1.1	Обзор задач, предлагаемых в школьном курсе. 1. Особенности становления обобщенного умения по решению задач из курса информатики. 2. Олимпиадные задачи по информатике. 3. Понятие задачи и подзадачи. Сведение задачи к подзадачам.	2					Устный опрос
2.1.2	Способы решения олимпиадных задач по информатике.		2			4	Проверка лабораторной работы

2.1.3	Итерационные методы: решето Эратосфена, схема Горнера, числа Фибоначчи, числовые ряды и т.д.		2			2	Проверка лабораторной работы
2.1.4	Рекурсия и рекуррентные соотношения. Динамическое программирование. Методы динамического программирования при решении олимпиадных задач по информатике.		2			4	Проверка лабораторной работы
2.2	Задачи обработки чисел		6			8	
2.2.1	Позиционные и непозиционные системы счисления. Перевод чисел из одной позиционной системы счисления в другую. Кодирование различных видов информации.		2			2	Проверка лабораторной работы
2.2.2	Задачи целочисленной арифметики. Выделение цифр числа двумя способами. Операции с дробями, сокращение дробей. Цепные дроби. Разложение рациональных и иррациональных чисел в цепные дроби.		2			4	Проверка лабораторной работы
2.2.3	Действия с многозначными числами: сложение, вычитание и произведение многозначных чисел.		2			2	Проверка лабораторной работы
2.3	Поиск и сортировка		6			4	
2.3.1	Произвольный поиск в произвольно организованных структурах.		2				Проверка лабораторной работы
2.3.2	Поиск в упорядоченно организованных структурах. Сокращение областей поиска.		2			2	Проверка лабораторной работы
2.3.3	Методы сортировки: простой вставкой, бинарной вставкой, слияниями (алгоритм фон Немана) и др.		2			2	Проверка лабораторной работы
2.4	Задачи комбинаторики. Графы	2	4			10	
2.4.1	Комбинаторика, понятие об упорядочении и группировании объектов. Виды соединений (комбинаций): перестановки, сочетания, размещения. Соединения с повторениями. Реализация перебора вариантов.	2				2	Устный опрос
2.4.2	Использование видов соединений при решении задач повышенной сложности по информатике.		2			4	Проверка лабораторной работы
2.4.3	Графы. Алгоритмы решения задач на графах.		2			4	Проверка лабораторной работы

						ной работы,
3.	Решение практикоориентированных задач. Творческий проект: постановка задачи, проектирование и реализация	2	12			12
3.1	Творческий проект: постановка <u>практикоориентированной</u> задачи. Постановка задачи, представленной на естественном языке, с учетом конкретной предметной области. Системный анализ задачи.		2			Проверка лабораторной работы
3.2	Использование современных технологий и средств программирования для обработки и визуализация данных	2	6			10
3.2.1	Геометрические задачи. Представление геометрических объектов на плоскости: точка, отрезок, прямая. Взаимное расположение геометрических объектов.		2			Проверка лабораторной работы
3.2.2	Треугольники. Многоугольники. Площади фигур. Взаимное размещение фигур на плоскости.		2			2
3.2.3	Визуализация объектов в современных средах программирования при решении задач из разных предметных областей.	2				4
3.2.4	Моделирование и визуализация физических законов, процессов и явлений.		2			4
3.3	Творческий проект: верификация результатов, защита		4			2
3.3.1	Тестирование проекта. Оценка эффективности и результативности работы программного средства.		2			2
3.3.2.	Защита проекта.		2			Защита проекта
	Итого: 42 ч.	8	34			44

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература:

1. Задачи по программированию / С.М. Окулов, Т.В. Ашихмина, Н.А. Бушмелева [и др.]; под. ред. С.М. Окулова. – Москва: Бином, 2006.
2. Котов, В.М. Информатика. Методы алгоритмизации: учеб. пособ. для 8 – 9 кл. общеобразоват. шк. с углубл. изучением информатики / В.М. Котов, И.А. Волков, А.И. Лапо. – Минск: Народная асвета, 2000.
3. Котов, В.М. Информатика. Методы алгоритмизации: учеб. пособ. для 10-11 кл. общеобразоват. шк. с углубл. изучением информатики / В.М. Котов, И.А. Волков, А.И. Лапо. – Минск: Народная асвета, 2000.
4. Котов, В.М. Структуры данных и алгоритмы: теория и практика / В.М. Котов, Е. П. Соболевская. – Минск: БГУ, 2004.
5. Меньшиков, Ф.В. Олимпиадные задачи по программированию. – Санкт-Петербург: Питер, 2006.
6. Окулов, С.М. Программирование в алгоритмах / С.М. Окулов. – Москва: БИНОМ, 2002.

Дополнительная литература:

7. Босова, Л.Л. Занимательные задачи по информатике / Л.Л. Босова. – Москва: Бином, 2005.
8. Сафронов, И. Задачник-практикум по информатике / И. Сафронов. – Санкт-Петербург: БХВ, 2002.
9. Семакин, И.Г. Информатика: задачник-практикум (7 – 11 классы) / И.Г. Семакин, Е.К. Хеннер. – Москва: Бином, 2005.
10. Угринович, Н. Практикум по информатике и информационным технологиям / Н. Угринович, Л. Босова, Н. Михайлова. – Москва: ЛБЗ, 2007.
11. Кормэн, Т.Х. Алгоритмы: построение и анализ / Т.Х. Кормэн, Ч.И. Лейзерсон, Р.Л. Ривест. – Москва: МЦНМО, 2000.

**Примерный ТЕМАТИЧЕСКИЙ ПЛАН
ДЛЯ СПЕЦИАЛЬНОСТИ 1-02 05 02 ФИЗИКА И ИНФОРМАТИКА**

№	Наименование разделов, тем	Всего	Лекции	Лабораторные
	Раздел 1. Процесс решения задач с помощью компьютера	2	2	
1.1	Этапы решения задач с помощью компьютера	2	2	
	Раздел 2. Методы и способы решения задач по информатике	26	4	22
2.1	Общая характеристика методов и способов решения задач по информатике	8	2	6
2.2	Задачи обработки чисел	6		6
2.3	Поиск и сортировка	6		6
2.4	Задачи комбинаторики. Графы	6	2	4
	Раздел 3. Решение практикоориентированных задач. Творческий проект: постановка задачи, проектирование и реализация	14	2	12
3.1	Тема 3.1. Творческий проект: постановка практикоориентированной задачи	2		2
3.2	<i>Тема 3.2. Использование современных технологий и средств программирования для обработки и визуализация данных</i>	8	2	6
3.3	Тема 3.3. Творческий проект: верификация результатов, защита	4		4
	Итого:	42	8	34

Материалы на электронных носителях

(сайт физико-математического факультета, локальная сеть физико-математического факультета, кафедральные компьютеры)

1. Тексты лекций по учебной дисциплине.
2. Задания для самостоятельной работы.
3. Вопросы к экзамену.
4. Типовые задания к экзамену.

Методические рекомендации по организации и выполнению самостоятельной работы студентов

Содержание и формы самостоятельной работы студентов разрабатываются кафедрами в соответствии с целями и задачами подготовки специалиста. Для управления самостоятельной работой рекомендуется использовать электронные средства обучения, тестирующие программы. Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

Перечень используемых средств диагностики результатов учебной деятельности

Основным средством диагностики усвоения знаний, умений и овладения необходимыми навыками по учебной дисциплине являются:

– *фронтальный опрос* на лекционных занятиях, направлен на систематизацию знаний студентов, определение уровня готовности аудитории к восприятию нового материала, а также на формирование у преподавателя представление об усвоении студентами основополагающих понятий и фактов изучаемой учебной дисциплины;

– *проверка заданий* разнообразного типа (рецептивные, репродуктивные, продуктивные, творческие), выполняемых в рамках часов, отводимых на учебные занятия (лабораторные), представляет собой диагностику систематичности подготовки студентов к занятиям, уровень усвоения ими практико-ориентированного содержания программного материала учебной дисциплины;

– *групповые и индивидуальные консультации студентов* предназначены для диагностики уровня овладения определенными знаниями, умениями и навыками, как теоретического материала, так и практического; устранения типичных ошибок и пробелов в знаниях обучающихся;

– *самостоятельные работы* используются для определения индивидуальных особенностей, темпа продвижения студентов и усвоения ими необходимых знаний;

– *компьютерное тестирование* позволяет относительно быстро провести диагностику усвоения студентами учебного материала как по

отдельным темам и разделам учебной дисциплины, так и по учебной дисциплине в целом;

– *зачет* используется для осуществления итоговой диагностики усвоения учащимися содержания учебной дисциплины за учебный семестр и оценивается обычно в форме «зачтено» или «не зачтено» в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования по десятибалльной шкале.

РЕПОЗИТОРИЙ БГПУ

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Методика преподавания физики	Кафедра физики и методики преподавания физики	При рассмотрении вопросов связанных с решением практикоориентированных задач (Раздел 3) использовать согласованную терминологию в соответствии с действующими учебными пособиями для учреждений общего среднего образования.	Протокол № 5 от 23.12.2015