

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«САНКТ-ПЕТЕРБУРГСКИЙ  
ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ ПРОФСОЮЗОВ»**

Кафедра Информатики и математики  
(полное наименование кафедры)

УТВЕРЖДЕН  
на заседании кафедры

Протокол №1 от 01.06.2020

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ**

**Информатика и программирование**  
(наименование дисциплины)

**09.03.03 «Прикладная информатика»**  
(код наименования направления подготовки /специальности/)

**Прикладная информатика в экономике**  
(направленность/профиль/)

## **1. Общие положения**

Фонд оценочных средств (ФОС) по дисциплине используется в целях нормирования процедуры оценивания качества подготовки и осуществляет установление соответствия учебных достижений запланированным результатам обучения и требованиям образовательной программы дисциплины. Предметом оценивания являются знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций у обучающихся. Процедуры оценивания применяются в процессе обучения на каждом этапе формирования компетенций посредством определения для отдельных составных частей дисциплины методов контроля – оценочных средств. Основным механизмом оценки качества подготовки и формой контроля учебной работы студентов являются текущий контроль успеваемости и промежуточная аттестация.

### **1.1. Цель и задачи текущего контроля студентов по дисциплине**

Цель текущего контроля – систематическая проверка степени освоения программы 09.03.03 «Прикладная информатика» дисциплины «Информатика и программирование» уровня достижения планируемых результатов обучения - знаний, умений, навыков, в ходе ее изучения при проведении занятий, предусмотренных учебным планом. Задачи текущего контроля:

1. обнаружение и устранение пробелов в освоении учебной дисциплины;
2. своевременное выполнение корректирующих действий по содержанию и организации процесса обучения;
3. определение индивидуального учебного рейтинга студентов;
4. подготовка к промежуточной аттестации.

В течение семестра при изучении дисциплины реализуется традиционная система поэтапного оценивания уровня освоения. За каждый вид учебных действий студенты получают оценку.

### **1.2. Цель и задачи промежуточной аттестации студентов по дисциплине.**

Цель промежуточной аттестации – проверка степени усвоения студентами учебного материала, уровня достижения планируемых результатов обучения и сформированности компетенций на момент завершения изучения дисциплины. Промежуточная аттестация проходит в форме зачёта и экзамена.

Задачи промежуточной аттестации:

1. определение уровня освоения учебной дисциплины;
2. определение уровня достижения планируемых результатов обучения и сформированности компетенций;
3. соотнесение планируемых результатов обучения с планируемыми результатами освоения образовательной программы в рамках изученной дисциплины.

## 2. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Таблица 1.

№ п\п	Контролируемые темы дисциплины	Код формируемой компетенции	Код и наименование индикатора достижения	Наименование оценочного средства
1	Информация и информационные процессы	УК-1, ОПК-1	<p>УК -1.1. Знает основные положения информатики как научной дисциплины, изучающей процессы обработки информации, принципы организации и функционирование современных средств обработки информации, базовые принципы информационного анализа процессов и явлений предметной области.</p> <p>УК -1.2. Умеет соотносить разнообразные явления и систематизировать их в рамках избранных видов профессиональной деятельности.</p> <p>УК -1.3. Имеет практический опыт работы с информационными источниками, опыт научного поиска, создания научных текстов.</p> <p>ОПК-1.1. Знает основные положения информатики как научной дисциплины, изучающей процессы обработки информации, принципы организации и функционирование современных средств обработки информации, базовые принципы информационного анализа процессов и явлений предметной области; методы проектирования и разработки алгоритмов и программ</p> <p>ОПК-1.3. Владеет навыками информационного анализа процессов и явлений, составления алгоритмов.</p>	Опрос, участие в коллоквиуме, выполнение практических заданий, кейсы
2	Программные средства обработки информации	ОПК-1, ОПК-4	<p>ОПК-1.2. Умеет решать задачи обработки данных; составлять алгоритмы, проектировать, программировать программы на одном из языков высокого уровня с использованием современных технологий программирования</p> <p>ОПК-1.3. Владеет навыками информационного анализа процессов</p>	Подготовка сообщений, выполнение практических заданий, тестирование

			и явлений, составления алгоритмов. ОПК-4.1. Знает основные положения информатики как научной дисциплины, изучающей процессы обработки информации, принципы организации и функционирование современных средств обработки информации, базовые принципы информационного анализа процессов и явлений предметной области; методы проектирования и разработки алгоритмов и программ.	
3	Постановка и алгоритмизация прикладных задач	ОПК-4, ОПК-7	ОПК-4.2. Умеет решать задачи обработки данных с помощью современных инструментальных средств конечного пользователя; составлять алгоритмы, проектировать, программировать и тестировать программы на одном из языков высокого уровня с использованием современных технологий программирования ОПК-4.3. Владеет навыками документирования программ на одном из языков высокого уровня с использованием современных программных средств ОПК-7.1. Знает принципы организации проектирования и содержание этапов процесса разработки программных комплексов дисциплины, задачи и методы исследования и обеспечения качества и надежности программных компонентов, основы разработки программных комплексов. ОПК-7.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.	Опрос, тестирование, коллоквиум, кейсовые задания
4	Методы и языки программирования	ОПК-4, ОПК-7	ОПК-4.2. Умеет решать задачи обработки данных с помощью современных инструментальных средств конечного пользователя; составлять алгоритмы, проектировать, программировать и тестировать программы на одном из языков высокого уровня с использо-	Опрос, тестирование, коллоквиум, практические задания

			<p>ванием современных технологий программирования</p> <p>ОПК-4.3. Владеет навыками документирования программ на одном из языков высокого уровня с использованием современных программных средств</p> <p>ОПК-7.1. Знает принципы организации проектирования и содержание этапов процесса разработки программных комплексов дисциплины, задачи и методы исследования и обеспечения качества и надежности программных компонентов, основы разработки программных комплексов.</p> <p>ОПК-7.2. Умеет проводить анализ и проектирование современного программного обеспечения, определять его качественные показатели, организовывать процесс разработки и вести документацию в соответствии с современными стандартами.</p> <p>ОПК-7.3. Владеет навыками программирования, отладки и тестирования программного обеспечения.</p>	
Результат достижения планируемых результатов изучения дисциплины			зачёт, экзамен	

### 3. Описание показателей и критериев оценивания компетенций

#### 3.1. Критерии оценивания (текущий контроль)

1. Оценка «отлично» выставляется студенту, если студент имеет глубокие знания учебного материала по теме практического задания, в логической последовательности излагает материал; смог ответить на все уточняющие и дополнительные вопросы;
2. Оценка «хорошо» выставляется, если студент показал знание учебного материала, смог ответить почти полностью на все заданные дополнительные и уточняющие вопросы;
3. Оценка «удовлетворительно» выставляется, если студент в целом освоил материал; однако, ответил не на все уточняющие и дополнительные вопросы;
4. Оценка «неудовлетворительно» выставляется студенту, если он имеет существенные пробелы в знаниях основного учебного материала по теме практического задания, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

#### 3.2. Критерии оценивания (зачет)

Знания, умения, навыки и компетенции студентов оцениваются следующими оценками: «зачтено», «не зачтено».

«Зачтено» выставляется студенту при условии, что студент твердо знает программный материал, грамотно и последовательно его излагает, увязывает с практикой, владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

«Не зачтено» выставляется студенту при условии, что студент не знает значительной части основного программного материала, в ответе допускает существенные ошибки, неправильные формулировки, не владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

### 3.3. Критерии оценивания (экзамен)

Знания, умения, навыки и компетенции студентов оцениваются следующими оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» (Таблица 2.).

Таблица 1.

Оценка	Критерии оценивания
Отлично	Студент не только глубоко и прочно усвоил весь программный материал, но и проявил знания, выходящие за его пределы, почерпнутые из дополнительных источников (классическая литература, учебная литература, научно-популярная литература, научные статьи и монографии и т. п.); умеет самостоятельно обобщать программный материал, не допуская ошибок, проанализировать его с точки зрения различных школ и взглядов; увязывает знания с практикой, приводит примеры, демонстрирующие глубокое понимание материала или проблемы, свободно справляется с задачами и практическими заданиями; исчерпывающе, последовательно, грамотно и логически стройно выстраивает свой ответ.
Хорошо	Студент твердо знает программный материал, грамотно и последовательно его излагает, увязывает с практикой, не допускает существенных неточностей в ответе на вопросы, может правильно применять теоретические положения и владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач, испытывает незначительные затруднения при самостоятельном обобщении программного материала.
Удовлетворительно	Студент усвоил только основной программный материал, но не знает его отдельных положений, в ответе допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала, не в полной мере владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач, испытывает затруднения при самостоя-

	тельном обобщении программного материала.
Неудовлетворительно	Студент не знает значительной части основного программного материала, в ответе допускает существенные ошибки, неправильные формулировки, не владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач, испытывает значительные затруднения при самостоятельном обобщении программного материала.

### *ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИ*

- 1. Вопросы для подготовки к промежуточной аттестации по дисциплине (зачету)**
2. Информация как ресурс. Способы хранения и обработки информации. Этические и этические конфликты в информационном обществе. Информационно-культурное пространство интернета и виртуальной реальности.
3. Понятие структурированных данных. Определение и назначение базы данных. Соблюдение кодексов профессиональной этики при использовании информационных ресурсов сети интернет.
4. Концепция информационного общества. Современные проблемы информатизации общества. Информационные процессы. Информационные системы. Этапы развития, примеры, классификации и структура информационных систем.
5. Современные информационные технологии. Понятия, основные определения, сущность информационных технологий. Сетевая коммуникация и распространение визуальной информации.
6. Содержательный и технологический аспекты процесса интернетизации. Ключевые тенденции развития медиакоммуникаций и социальные сети. Риски, связанные с использованием Интернет-пространства.
7. Взаимодействие гуманитарной и технологической цивилизаций. Основные причины формирования единого информационного пространства.
8. Назначение, основные возможности текстового редактора Word. Запуск и завершение работы. Основные элементы окна. Справочная система: команды, окна, Мастер подсказок (помощник), контекстно-зависимая справка. Режимы просмотра документов.
9. Управление окном приложения и окном документа. Структура меню и кнопочных панелей инструментов текстового процессора. Настройка панелей инструментов и лент. Колонтитулы и сноски: назначение, виды, способы создания.
10. Понятия текстового файла и текстового документа. Создание текстовых файлов и текстовых документов. Основные элементы текстового документа: символы, слова, строки, предложения, абзацы. Структура страницы. Структура документа.
11. Операции форматирования символов и абзацев. Общие правила форматирования символов и абзацев. Копирование формата.
12. Основные операции с текстом. Перемещение по набранному тексту. Способы выделения объектов текстового документа. Поиск и замена. Удаление, перемещение и копирование объектов текстового документа. Вставка полного текста из другого файла. Вставка специальных символов из таблицы символов.
13. Понятие шрифта. Параметры шрифтов. Шрифты TrueType. Установка стандартных гарнитуры и кегля по умолчанию в шаблоне.
14. Основные элементы страницы текстового документа Word. Создание и редактирование колонтитулов. Вставка сносок, примечаний. Блокировка разрыва страницы между абзацами. Подгонка страницы. Вставка номеров страниц, даты создания документа. Вставка разрывов страницы.
15. Работа с таблицами. Способы создания таблиц Выделение структурных элементов. Основные операции с таблицами: объявление первой строки таблицы заголовком; пере-

мещение по ячейкам таблицы; вставка и удаление строк, столбцов; изменение ширины (высоты) столбца (строки). Форматирование таблицы.

16. Преобразование текстового документа в таблицу. Обрамление и заливка выделенных ячеек таблицы. Сохранение таблицы как элемента автотекста. Управление расположением таблицы на странице документа. Сортировка содержимого одного или нескольких столбцов.
17. Арифметические операции над числовыми данными таблиц, использование формул. Построение диаграмм. Выбор типа диаграммы. Обрамление диаграммы рамкой. Изменение формата оси категорий и оси значений, формата оси сетки. Изменение названия диаграммы, оси значений, оси категорий.
18. Вставка рисунков. Способы связи рисунка с документом. Операции с рисунками: перемещение рисунка в другую позицию, масштабирование, изменение размера, создание рамки вокруг рисунка, редактирование. Использование графики в качестве подложки.
19. Понятие шаблона документа. Шаблоны документов, предоставляемые Word: типы, виды. Оформление документов с использованием готовых шаблонов. Создание собственных шаблонов на основе шаблона Обычный. Создание нового шаблона на основе текстового документа.
20. Списки простые и многоуровневые. Способы создания списков. Встроенные форматы списков. Изменение маркеров, создание произвольного символа маркера. Преобразование списков. Изменение положения маркера, положения текста относительно маркера страниц. Формат номера страницы.
21. Размещение в ячейках и использование данных разных типов в электронной таблице. Содержимое ячейки.
22. Основные элементы электронной таблицы. Понятие адреса, ссылки. Типы ссылок. Вычисления по формулам и с помощью стандартных функций в электронных таблицах.
23. Графическое представление табличных данных. Типы диаграмм. Понятия рядов данных и категорий значений. Элементы оформления диаграмм.
24. Редактирование диаграмм. Форматирование типов диаграмм, элементов диаграммы, области диаграммы и области построения диаграммы, рядов данных, осей и сетки.
25. Использование логических функций и функций для работы с базой данных в электронных таблицах. Сортировка данных в электронных таблицах.
26. Понятие фильтрации данных в списке. Команда фильтрации, Автофильтр, расширенный фильтр.
27. Информационный подход к анализу процессов и явлений. Основные стадии анализа. Пример информационного анализа.
28. Качественные и количественные факторы. Оцифровка и нормализация значений факторов.

### **Вопросы для подготовки к промежуточной аттестации по дисциплине (экзамену)**

1. Информатика как фундаментальная наука и область практической деятельности. Предмет научной информатики. Общекультурная роль информатики. Понятие об информационном обществе.
2. Информация как базовое понятие науки. Информационные системы и технологии. Этапы развития информационных технологий.
3. Фазы информационного процесса для получения информации. Этапы преобразования информации. Данные и знания.
4. Синтаксический аспект информации. Определение количества информации по Шеннону. Формулы кодирования и Хартли, примеры их применения.
5. Семантический и прагматический аспекты информации.
6. Представление дискретной информации в компьютере. Принципы сжатия. Примеры представления текстовой и графической информации.



7. Представление непрерывной информации в компьютере. Емкость канала передачи. Примеры представления звуковой информации.
8. Информационный подход к анализу процессов и явлений. Основные стадии анализа. Пример информационного анализа.
29. Качественные и количественные факторы. Оцифровка и нормализация значений факторов.
30. Понятие программного обеспечения. Классификация программных средств. Структура системного программного обеспечения. Операционные системы и их характеристики.
31. Сервисное программное обеспечение. Системные утилиты. Программы резервного копирования. Архиваторы и примеры их применения.
32. Вирусы и антивирусное программное обеспечение. Защита от сетевых атак.
33. Классификация программных средств для обработки текстовых документов.
34. Варианты создания документа в текстовом редакторе. Редактирования текстовых документов: цели и средства.
35. Форматирования текстовых документов: цели и средства. Вставка специальных элементов в текстовый документ (сноски, примечания, аннотации, специальные символы и др.).
36. Создание и редактирование таблиц в текстовом документе.
37. Вставка в текстовый документ оглавления, указателей, списка иллюстраций.
38. Операции с графическими объектами в текстовом редакторе.
39. Подготовка больших документов к печати (выделение разделов, нумерация страниц, формирование колонтитулов, предварительный просмотр).
40. Вычисления в текстовом процессоре; использование вычисляемых полей и формул.
41. Настройка окна табличного редактора, режимы просмотра документа. Навигация в текстовых документах с помощью закладок, переходов, гиперссылок.
42. Размещение в ячейках и использование данных разных типов в электронной таблице.
43. Цели и средства форматирования элементов электронных таблиц. Создание и применение пользовательских форматов.
44. Использование последовательностей и автозаполнения в электронных таблицах.
45. Вычисления по формулам и с помощью стандартных функций в электронных таблицах.
46. Графическое представление табличных данных. Типы диаграмм. Понятия рядов данных и категорий значений. Элементы оформления диаграмм.
47. Сортировка данных в электронных таблицах.
48. Работа с электронной таблицей как с базой данных.
49. Использование логических функций и функций для работы с базой данных в электронных таблицах.
50. Определения понятия задача. Фазы постановки задач.
51. Основные стадии компьютерного решения прикладной задачи.
52. Общее понятие алгоритма. Свойства компьютерных алгоритмов.
53. Теория алгоритмов.
54. Машина Тьюринга.
55. Теория вычислительной сложности.
56. Методика исследования вычислительной сложности задачи.
57. Способы представления алгоритмов.
58. Блок-схемы.
59. Основные алгоритмические конструкции: следование, цикл, переход, ветвление.
60. Стандартные алгоритмы.
61. Язык программирования как средство представления алгоритма.
62. Классификация языков программирования. Подходы к выбору языка. Версии языка.
63. Содержание и порядок выполнения отдельных этапов создания программы. Компоненты и возможности системы программирования. Модули системы.
64. Понятие типов данных в языке программирования.

65. Основные компоненты языка программирования: алфавит, словарь, набор операций, выражения, операторы, синтаксис, семантика.
66. Структура программы на языке Паскаль. Назначение и оформление раздела описания данных.
67. Описание одномерных и двумерных массивов. Размещение массивов в памяти.
68. Назначение процедур. Правила описания процедур и обращения к ним.
69. Присваивание значений переменным. Структура оператора присваивания. Понятие совместимости типов.
70. Операторы перехода и ветвления. Структура и синтаксис полного и неполного условного оператора.
71. Понятие заголовка и тела цикла. Варианты и синтаксис операторов цикла.
72. Применение, структура и синтаксис и вложенных циклов.
73. Работа с одномерными и двумерными массивами в циклах.
74. Варианты и синтаксис операторов ввода и вывода информации.

#### **4. Типовые контрольные задания (тесты, рефераты, курсовые работы, кейсы и др.) и методические материалы, процедуры оценивания знаний, умений и навыков ТЕКУЩИЙ КОНТРОЛЬ**

##### **Методические рекомендации по написанию контрольных работ**

Контрольная работа (реферат) – это письменная работа на определенную тему, подготовленная на основе изучения учебной и научной литературы, статей, а также обобщения личных знаний и практического опыта. Контрольная работа по курсу «Информатика и программирование» представляет собой одну из форм самостоятельной работы студентов по изучению и усвоению наиболее сложных теоретических проблем. При написании контрольной работы студенты должны показать умение работать с основной и дополнительной литературой, способность обобщать материал.

Основные требования к реферату (контрольной работе):

- высокий теоретический уровень;
- четкость определения понятий и категорий;
- выявление правовых аспектов рассматриваемой проблемы;
- аргументированность основных положений и выводов работы;
- обязательное использование фактического (в том числе статистического) материала;

Выбор темы – одна из наиболее ответственных составных частей разработки контрольной работы. Любая тема по курсу «Документирование управленческой деятельности», особенно в условиях современного реформирования экономики России, является актуальной. Следовательно, знания, которые студент приобретет при написании контрольной работы, принесут положительные результаты, как с научной, так и с практической точек зрения. Не исключено, что в процессе работы название контрольной работы в рамках избранной темы потребует уточнения. Но, в конечном счете, нужно соблюдать главное требование: название темы должно соответствовать общему направлению излагаемого материала, отражать его основные идеи.

Студент может предложить свою тему контрольной работы, если она соответствует тематике курса. В этом случае тему необходимо согласовать с преподавателем, ведущим данный курс.

План – это основа, костяк письменной работы. От правильного его составления во многом зависят структура, содержание, логическая связь ее частей. Одновременно план – важный показатель того, насколько глубоко студент изучил экономические произведения (первоисточники) и другой необходимый материал (дополнительную литературу), сумел ли он отобрать из всего этого главное, существенное. Разработка плана состоит из двух этапов: составления предварительного плана и составления окончательного плана. Составление оконча-

тельного плана осуществимо лишь на основе более или менее полного овладения материалом, уяснения, по крайней мере, главных вопросов темы.

В плане должны быть отражены: введение; 2-3 вопроса; заключение; список использованной литературы. Литература подбирается студентом самостоятельно. Кроме того, он может обратиться за рекомендациями к преподавателю, ведущему курс. Когда план составлен, студент приступает к глубокому изучению литературы с целью раскрытия содержания каждого из пунктов плана. В первую очередь изучаются первоисточники и основная литература по теме. Кроме того, необходимо использовать и дополнительную литературу (монографии, брошюры, журнальные статьи и т.п.).

Работа над текстом – это кропотливый, систематический труд студента. Написание хорошей контрольной работы требует от автора подлинно добросовестного отношения к учебе, принципиальности, честности. Как и любая творческая работа, написание реферата (контрольной работы) – это длительное размышление над избранной темой, терпеливая переделка первого варианта, неустанное его совершенствование.

Написание следует начинать с введения. Именно во введении студент дает общую характеристику самой темы, краткий обзор изученной им литературы, показывает степень научной разработки проблемы, цели, а также предполагаемый личный вклад в ее достижение. В этом случае автор, видя целое, более целеустремленно и квалифицированно работает над темой.

Поскольку объем реферата (контрольной работы) фиксируется определенным количеством страниц, то перед написанием текста их необходимо распределить по главам или параграфам, с учетом введения и заключения, определить сроки выполнения каждой из глав. Разумеется, такое планирование может быть только ориентировочным, но в нем заложено дисциплинирующее и мобилизующее начало.

Следует заранее подготовить себя к тому, что написанный текст при редактировании потребует немалых доработок и переделок, а зачастую и переписывания заново целых страниц и даже глав. Таким образом, первоначальный вариант текста может оказаться лишь черновиком. Если хочешь научиться хорошо писать, научись хорошо думать! Неточное изложение является показателем того, что предмет мышления и самому автору недостаточно ясен. В свою очередь, правильное мышление зависит от знания предмета исследования. В конце каждой главы или параграфа целесообразно сделать небольшие выводы по наиболее важным аспектам главы. Выводы помогают плавно перейти к следующему вопросу. Формулировка выводов, а также предложений и рекомендаций по главам или параграфам не отменяет необходимости заключения по работе в целом. По существу, такое заключение – особая глава реферата (контрольной работы), в которой в обобщенном виде подводятся итоги всего исследования, формулируются общие выводы, авторские предложения, отмечаются вопросы, которые не нашли в исследовании окончательного решения и требуют дальнейшей разработки. Выводы должны быть отработаны так, чтобы словам было тесно, а мыслям просторно. Выводы декларативного характера, не вытекающие из содержания работы, снижают ее уровень.

Реферат (контрольная работа), представляя собой уже законченную текстуально работу, нуждается в обязательном авторском редактировании. Речь идет об упорядочении текста в соответствии с требованиями и правилами литературного языка и стиля, характером, назначением и направленностью проведенного исследования. Редактирование требует от автора: исключительного внимания и сосредоточенности; окончательного уяснения изучаемого вопроса; умения критически подойти к своему тексту; решительного устранения всего лишнего, мешающего четкому освещению вопроса (повторений, многословия, пространственных формулировок); осуществления его лишь после полного написания текста; скромности (избегайте ссылок на самого себя и местоимений «я», «мною», «мой» и т.п.); уважения мнения других. В ходе творческих дискуссий необходимо придерживаться максимально аргументированных высказываний; научной ответственности и добросовестности.

Структурно контрольная работа состоит из следующих элементов:

1. Введение (актуальность темы и основные задачи работы).
2. Основная часть (состоит из двух – трех вопросов; в первом параграфе рассматриваются теоретические аспекты проблемы, выводятся определения понятий, дается общая характе-

ристика состояния того или иного явления в современной экономической действительности; во втором параграфе анализируются перспективы, прогнозируется динамика рассматриваемых хозяйственных процессов, обосновываются узловые положения).

3. Заключение (формулируются основные выводы и определяется конкретное их значение для практики хозяйственных преобразований в современной российской экономике);

4. Список использованной литературы (перечень основной и дополнительной литературы).

Общий объем контрольной работы должен составлять 10-12 страниц компьютерного текста через 1,5 интервала. На втором листе контрольной работы (после титульного листа) приводится ее план. На последнем листе дается список использованной литературы в следующей последовательности: 1) документы: законы, указы, постановления правительства; 2) труды и выступления руководителей государства и правительства, деятелей законодательной и исполнительной властей; 3) монографии, брошюры, статьи; 4) статистические и справочные издания; 5) электронные источники.

Список использованной литературы должен быть оформлен в соответствии с требованиями государственного стандарта:

### ***Представление контрольной работы***

Выполненная работа сдается методистам заочного факультета университета в установленный учебным планом срок, где она регистрируется и передается на проверку преподавателю. После проверки контрольная работа возвращается студенту для ознакомления замечаниями и рецензией. В случае незачета работы студент должен внести в нее все необходимые исправления и дополнения, вытекающие из замечаний преподавателя. Проверенная работа (со всеми исправлениями, дополнениями и первой рецензией) представляется студентом на повторную проверку.

### **Задания для написания контрольных работ (для заочной формы обучения).**

Общее задание — освоить способы моделирования и анализа экономических задач средствами электронных таблиц, работу с электронной таблицей как с базой данных, стандартные функции для работы с базой данных, применение расширенного фильтра и итогов. Самостоятельно смоделировать и проанализировать задачу из индивидуального задания.

- База данных «Программное обеспечение предприятия».
- База данных «Техническое обеспечение предприятия».
- База данных «Оргтехника предприятия».
- База данных «Персонал».
- База данных «Интернет ресурсы по информатике».
- База данных «Интернет ресурсы по программированию».
- База данных «Интернет ресурсы по языку программирования».
- База данных «Интернет магазины программного обеспечения».
- База данных «Интернет магазины мультимедиа контента».

**Требования к оформлению контрольной работы** подробно представлены в Положении о бюро контрольных работ, размещенном на сайте Университета в личном кабинете на странице в Системе поддержки самостоятельной работы студентов **ПОЛОЖЕНИЕ О БЮРО КОНТРОЛЬНЫХ РАБОТ \_ для работ студентов заочной формы обучения.**

### **4. Методические рекомендации по написанию курсовой работы**

Целью курсовой работы является получение студентами практических навыков в решении задач с применением компьютера. При выполнении курсовой работы студенты должны изучить и практически освоить правила и приемы:

- перехода от содержательной формулировки задачи к постановке ее в виде, учитывающем применение компьютера;
- разработки алгоритма решения задачи и представления его в виде графических блок-схем;
- представления алгоритма в виде программы на определенном алгоритмическом языке;
- подготовки материалов, необходимых для отладки программы на компьютер;
- отладки и тестирования программы;
- оформления результатов решения и разработки технической документации к программе.

При выполнении курсовой работы студент должен разработать программу решения сформулированной в задании задачи, подготовить материалы для отладки и тестирования программы на компьютере, отладить программу и оформить отчетную документацию по выполненной работе.

Методическое обеспечение процесса выполнения курсовой работы представлено в электронном учебно-методическом комплексе университета и включает полнотекстовые учебники и справочники по языку программирования Паскаль, более 80 индивидуальных заданий на курсовую работу и пример оформления пояснительной записки

## **1. Требования к разрабатываемым программам**

Разрабатываемая при выполнении курсовой работы программа решения задачи, сформулированной в задании, должна удовлетворять требованиям, приведенным ниже.

1. В программе должен быть предусмотрен ввод значений исходных данных из файла и вывод результатов в файл.
2. Форма ввода исходных данных должна быть удобной с точки зрения задания их пользователем, должна учитывать особенности задачи и обеспечивать максимально простое изменение отдельных значений.
3. В программе должен быть предусмотрен контрольный вывод исходных данных, следующий непосредственно за их вводом.
4. Должны быть предусмотрены контроль корректности введенных исходных данных и вывод сообщений при обнаружении некорректных значений.
5. Все предусмотренные в программе "аварийные" завершения ее работы (в случае возникновения исключительных ситуаций) должны сопровождаться выводом поясняющих сообщений.
6. В ходе работы программы должны выводиться промежуточные результаты, получаемые после выполнения каждого логически завершенного ее участка.
7. Вывод в файл группы значений однородных данных должен производиться после формирования выходного массива.
8. Форма вывода результатов в выходной файл должна быть удобной для восприятия и содержать текстовые пояснения в терминах решаемой задачи, заголовки и т.д.
9. Схемы алгоритмов и тексты программ должны быть разработаны с учетом принципов структурного программирования.

## **2. Оформление курсовой работы**

Результаты выполнения курсовой работы оформляются в виде пояснительной записки, в которой должны быть отражены результаты работы, проделанной на каждом этапе разработки и отладки программы.

Пояснительная записка подготавливается в текстовом редакторе Word и распечатывается на стандартных листах бумаги формата А4 (210x297). Схемы, рисунки, чертежи выполняются с использованием соответствующих автофигур на листах бумаги того же формата. Допускается применение специализированных программных средств для создания схем, например MS Visio и др. Подобные фрагменты должны быть корректно импортированы в файл отчета – так, чтобы для их отображения не требовалась установка соответствующих программ.

Схемы алгоритмов оформляются в соответствии с ГОСТ 19.701-90. Весь материал пояснительной записки должен быть сброшюрован и снабжен титульным листом установленного в СПбГУП образца, с указанием номера задания, номера группы, фамилии студента и фамилии руководителя курсовой работы.

### **3. Вопросы, требующие согласования с руководителем курсовой работы**

Выполнение курсовой работы должно проводиться в тесном контакте с руководителем курсового проектирования на практических занятиях и консультациях, где должны быть решены все возникающие вопросы и по отдельным этапам, и по всей работе в целом. Ряд вопросов, от решения которых в наибольшей степени зависит успешное выполнение работы, или в значительной степени влияющих на расход вычислительных ресурсов при разработке и запуске программ, требуют обязательного согласования с руководителем. К этим вопросам относятся:

- 1) формулировка конечной цели, которая должна быть достигнута при решении задачи, и пример решения, иллюстрирующий понимание задачи студентом;
- 2) перечень исключительных ситуаций, которые могут возникнуть при решении задачи на компьютер;
- 3) вид носителя и форма задания исходных данных;
- 4) варианты значений исходных данных, предназначенные для отладки и тестирования программы;
- 5) форма представления результатов работы программы.

### **4. Порядок выполнения курсовой работы**

Работа над созданием программы начинается с момента получения задания и включает в себя следующие основные этапы:

- 1) содержательная постановка задачи (составление технического задания);
- 2) формализованная постановка задачи (разработка математической модели);
- 3) алгоритмизация;
- 4) разработка структуры данных;
- 5) программирование и отладка;
- 6) тестирование программы;

Ниже приведены назначение и содержание каждого этапа, а также рекомендации по их выполнению.

#### **Этап 1. Содержательная постановка задачи**

Исходная задача формулируется, как правило, специалистами в соответствующей предметной области. Поэтому перед разработкой программы программисту необходимо добиться полного понимания содержания задачи. При этом должны быть получены ответы на следующие вопросы:

- понятна ли используемая терминология и сущность задачи (что должна делать программа);
- какие объекты, являются исходными для достижения цели (что дано), все ли исходные данные имеются, все ли данные необходимы, нет ли противоречий;
- какие объекты являются результатами (что должно быть получено);
- какие сделаны допущения;
- какие ограничения на область применения программы;
- какие дополнительные требования к программе должны быть выполнены (временные характеристики, требуемый объем памяти, точность представления исходных данных и результатов, виды носителей данных, защита информации и т.д.).

Итоги выполнения данного этапа должны быть проиллюстрированы на примере, отражающем сущность и особенности решаемой задачи. Полезно сопроводить пример иллюстрациями (графиками, рисунками, схемами и т.п.).

### **Этап 2. Формализованная постановка задачи**

Основным содержанием данного этапа является формализованное описание решаемой задачи и метода ее решения с помощью соответствующего математического аппарата.

Прежде, чем приступить к программированию, необходимо изучить или разработать методы решения поставленной задачи.

Как правило, для решения одной задачи существует несколько методов. Надо попытаться найти все или хотя бы некоторые из них, чтобы выбрать тот, который наиболее полно удовлетворяет требованиям к результатам и к программе. Для значительного числа задач может не существовать быстрого метода нахождения точного решения. Переборные методы допустимы только для задач ограниченной размерности. Возможность нахождения приближенных решений следует оговорить с руководителем.

При изучении уже существующего метода следует обратить внимание на следующие его особенности:

- расчетные формулы;
- характеристики метода по скорости, точности, требуемой памяти и области его применимости.

Исходя из того, что важнее для конкретного случая: скорость, точность или память, выбирается подходящий метод.

Выбрав (разработав) метод решения задачи, следует переформулировать ее в терминах соответствующего математического аппарата:

- определить и описать математическую форму представления исходных данных и конечных результатов;
- сформулировать метод решения задачи, т.е. выбрать необходимые преобразования, численные методы, формулы, определяющие правила получения результатов по исходным данным.

### **Этап 3. Алгоритмизация**

Алгоритмизация - это разработка алгоритма решения задачи, т.е. формулировка точного предписания, по которому должен осуществляться вычислительный процесс.

Для получения алгоритма необходимо выполнить следующие действия:

- 1) Преобразование формул к виду, удобному для алгоритмизации.

2) Проектирование схемы алгоритма. Алгоритм перед программированием целесообразно записать в виде псевдокода или блок-схемы.

Схемы должны отражать процесс разработки алгоритма методом пошаговой детализации с учетом принципов структурного программирования.

Схема первого (самого высокого) уровня составляется из укрупненных блоков, каждый из которых изображает типовой процесс обработки данных на компьютер (например, блок ввода данных, блок проверки правильности ввода данных, блок вычисления значений функции и т.п.). Каждому следующему уровню детализации алгоритма должна соответствовать схема, сопровождаемая указаниями на связь ее блоков с детализируемыми блоками предыдущего уровня.

Степень детализации на самом низком уровне должна быть такой, чтобы действия, соответствующие каждому блоку, могли быть легко записаны с помощью одного или небольшого числа предложений языка программирования.

В процессе детализации схемы алгоритма должны быть выделены участки программы, которые целесообразно оформить в виде отдельных подпрограмм (процедур или функций).

При разработке схемы алгоритма следует учитывать следующие правила:

- схема выполняется с учетом требований ГОСТ 19.701-90. На каждую подпрограмму и головную программу составляется отдельная схема;

- схема должна быть читаема без дополнительных пояснений автора, поэтому схемы алгоритмов должны быть снабжены достаточным для их понимания количеством комментариев;

- степень подробности схемы должна быть такой, чтобы она полностью размещалась на стандартном листе бумаги формата А4;

- управление в схеме должно передаваться в основном вниз, возвраты допускаются только в циклах. Альтернативно выполняемые ветви должны размещаться параллельно;

- схема алгоритма должна состоять только из основных управляющих структур типа "следование", "ветвление" и "цикл", для которых характерно наличие только одного входа и только одного выхода. Как следствие, каждая схема должна иметь только один блок "начало" и один блок "конец".

Обработка исключительных ситуаций. При составлении алгоритма следует предусмотреть поведение программы в исключительных ситуациях, т.е. ситуациях, в которых алгоритм может не давать результата. Такие ситуации могут возникать по следующим причинам:

- особенности конкретной задачи (например, система линейных уравнений не всегда имеет решение);

- ошибки во входных данных;

- нарушение пользователем установленного интерфейса.

Общим требованием к программам является то, что при любых исходных данных программа должна доработать до конца и выдать либо результат, либо сообщение о невозможности получить результат. Отсюда следует, что там, где может возникнуть исключительная ситуация, алгоритмом должны быть предусмотрены соответствующие проверки и действия по устранению или обходу этой ситуации.

#### **Этап 4. Разработка структуры данных**

На этом этапе конкретизируется представление данных в будущей программе.

Определяются структуры, типы и имена для объектов, используемых в программе с учетом диапазонов значений, требуемой точности представления исходных данных, промежуточных данных и результатов, а также с учетом особенностей используемого алгоритма.



Выбор имен является важным моментом в процессе решения задачи. Каждое имя должно в максимально возможной степени отражать назначение объекта и легко запоминаться. Следует избегать бессмысленных сокращений, использования трудно различимых комбинаций символов.

Необходимо учитывать, что структуры данных и алгоритмы оказывают друг на друга существенное влияние. В зависимости от выбора структуры данных программа может значительно меняться по размерам и времени выполнения. Выбор конкретного алгоритма во многих случаях определяет и требования к структуре данных. Поэтому структура данных уточняется и дополняется по мере детализации программы.

### **Этап 5. Программирование и отладка**

Основное содержание этапа - получение исходного текста программы.

Кодирование выполняется с обеспечением следующих требований.

Обязательно наличие комментариев - заголовков программы с указанием ее структурных и функциональных характеристик, правил обращения к программе, данных об авторе (фамилия, инициалы и номер группы студента), даты создания.

Делаются также комментарии к операторам объявления переменных (для каждой переменной в отдельности);

При оформлении текста программы используются отступы, сдвиги и пропуски при формировании текста. Операторы размещаются по строкам - не более одного оператора в строке.

Применяются правила структурного программирования, используются структурные программные конструкции ("следование", "ветвление", "цикл"). Выполняется автономная отладка модулей, затем комплексирование и отладка программы в целом.

При планировании отладки определяются точки программы, где будут использованы средства отладки, и принимаются решения о том, какие средства будут применены. Отладочные средства ставятся в узловых точках программы, на входах в процедуры; на длинных линейных участках ставят операторы для печати промежуточных результатов. Если есть возможность, используются различные отладочные режимы системы.

Результатом выполнения данного этапа является исполняемый модуль программы (файл с расширением .exe).

### **Этап 6. Тестирование программы**

Основное содержание - проверка соответствия программы требованиям, сформулированным в техническом задании.

В начале производится планирование процесса тестирования, разработка контрольных примеров (тестов).

Количество тестов должно быть таким, чтобы при испытаниях был обеспечен всесторонний контроль работы программы:

- проверка работы всех блоков не только для номинальных, но также для исключительных и предельных значений;
- проверка всех вариантов ввода и вывода, включая сообщения об ошибках;
- выполнение всех операторов и всех ветвей передачи управления.

Для тестирования методами белого ящика тесты сопровождаются таблицами, подтверждающими полноту тестирования по соответствующему методу, например в форме следующей матрицы, учитывающей покрытие тестами вариантов ветвлений в программном модуле.

Если результаты выполнения программы расходятся с ожидаемыми, то принимаются меры к поиску причин расхождения и устранению ошибок. Все внесенные изменения фикси-

руются вместе с указанием причины модификации программы. В данном случае требуются краткий перечень найденных на данном этапе ошибок.

## 5. Содержание пояснительной записки к курсовой работе

Пояснительная записка к курсовой работе должна содержать подробное описание процесса разработки и оформлена в соответствии с требованиями ГОСТ. Количество разделов курсовой работы должны совпадать с количеством этапов разработки программы. Разделам присваиваются заголовки, совпадающие с наименованиями этапов. Содержание каждого раздела должно отражать действия программиста и результаты, полученные на соответствующем этапе. Далее необходимо привести краткую характеристику программы, представляющая собой описание свойств программы, знание которых необходимо пользователю для ее эксплуатации. В это описание включаются: наименование программы; назначение программы; формат вызова программы; форматы ввода исходных данных и вывода результатов; простейшие метрики программы (количество строк исходного текста программы, размер исполняемого модуля, оценки времени работы программы).

Пояснительная записка должна заканчиваться заключением, содержащим выводы о проделанной работе и оценкой работоспособности полученной программы.

### Тестовые материалы ПАСПОРТ ТЕСТОВЫХ ЗАДАНИЙ

1. Общее количество тестовых заданий в базе – 212.
2. Ограничение времени выполнения теста (в минутах) – 30 минут.
3. Автоматическое перемешивание вопросов в тесте: - да (нет).
4. Случайный порядок ответов в тестовом задании: - да (нет).
5. Критерии оценки результатов тестирования:
  - Неудовлетворительно – 0 –55% правильных ответов.
  - Удовлетворительно -55 – 75% правильных ответов.
  - Хорошо – 75 -90% правильных ответов
  - Отлично – 90% и более правильных ответов

**Пример тестовых заданий для текущего контроля представлен ниже:**

1. Сведения (сообщения, данные) независимо от формы их представления – это
  - Документ
  - Публикация
  - База данных
  - **Информация**
2. Условием включения информации в информационные ресурсы является
  - Протоколирование
  - **Сохранение**
  - Документирование
  - Накопление
  -
3. Информацию, достаточную для решения тех или иных задач, называют
  - Понятной
  - Полезной
  - Актуальной
  - **Полной**
  - Достоверной
4. Информация в теории информации – это
  - **Сведения, полностью снимающие или уменьшающие существующую до их получения неопределенность**
  - Сведения, обладающие новизной

- Отраженное разнообразие
  - То, что поступает в наш мозг из многих источников и во многих формах и, взаимодействуя там, образует нашу структуру знания
  - Неотъемлемый атрибут материи
5. Информацию, не зависящую от чьего-либо мнения или суждения, называют
- Достоверной
  - Актуальной
  - **Объективной**
  - Понятной
  - Полезной
6. Кодом называется
- Правило, описывающее отображение набора знаков одного алфавита в набор знаков другого алфавита
  - Произвольная конечная последовательность знаков
  - **Правило, описывающее отображение одного набора знаков в другой набор знаков или слов**
  - Двоичное слово фиксированной длины
  - Последовательность слов над двоичным набором знаков
7. Русский алфавит может быть закодирован с помощью двоичных слов постоянной длины различными способами. Среди этих способов, отличающихся друг от друга, прежде всего, длиной кода, есть те, для которых длина кодового слова минимальна. Эта длина равна:
- 1
  - **6**
  - 5
  - 8
  - 2
8. В соответствии с кодовой таблицей ASCII символы английского алфавита кодируются двузначными числами, причем сочетание «I LOVE» кодируется так 733276798669, а сочетание «I LIVE»:
- **733276738669**
  - 733279768669
  - 733276867669
  - 733273768669
  - 763273737686
9. Укажите самое большое число
- $(756)_{13}$
  - $(756)_{12}$
  - **$(756)_{16}$**
  - $(756)_8$
  - $(756)_{10}$
10. Пиксель – это
- **Минимальный элемент растрового изображения**
  - Минимальный элемент векторного изображения
  - 32-битное изображение геометрическая точка
11. Растровое изображение представлено совокупностью
- Разноцветных линий
  - Графических примитивов
  - **Разноцветных точек**
  - Правильных треугольников

12. Ухудшение качества изображения, связанное с увеличением размеров, характерно для
- Растровой графики
  - Векторной графики
  - Автофигур, созданных в Word
13. Характерной особенностью векторной графики является
- Ухудшение качества изображения с уменьшением его размера
  - Ухудшение качества изображения с увеличением его размера
  - Уменьшение размера изображения с улучшением его качества
  - Неизменность качества изображения с увеличением его размера
14. Непечатаемые знаки
- Не печатаются на принтере
  - Не отображаются в документе после его сохранения
  - Показывают измененные места в документе
  - Показывают места в документе, которые не будут отпечатаны
15. Пробел ставится
- До знака препинания
  - До знака "дефис"
  - После знака "дефис"
  - После знака препинания
16. С помощью каких операций в текстовом редакторе Word можно в формуле выполнить умножение
- Использовать стандартную функцию PRODUCT
  - Использовать стандартную операцию \*
  - Использовать стандартную функцию УМНОЖ
  - Использовать стандартную функцию ROUND
  - Использовать стандартную функцию PRODUCT
17. Использовать стандартные ключевые слова можно использовать в WORD
- LEFT
  - CENTER
  - RIGHT
  - ABOVE
  - UNDER
18. Как начать список в текстовом редакторе Word
- Набрать 1. Иванов и нажать Enter
  - Набрать Иванов и нажать кнопку Нумерация
  - Выбрать стиль нумерованный список
  - Выбрать стиль номер строки
  - Набрать Иванов и выполнить **Формат - Список - Нумерованный - Ок,** набрать Иванов
19. Как закончить список в текстовом редакторе Word
- Два раза нажать Enter
  - Enter, затем Back Space
  - Enter, затем Delete
  - Enter и отключить список с помощью кнопки Нумерация
  - Enter, Формат - Список - Нет - Ок
20. Текстовая последовательность в электронной таблице Excel может быть использована при заполнении
- Названий месяца
  - Названий дней недели
  - Названий кварталов

- Любой текстовой последовательности, если для последовательности создан список пользователя
21. Числовая последовательность в электронной таблице Excel используется для заполнения интервала
- С любым шагом
  - С целыми числами
  - С дробными числами
  - С отрицательными числами
  - С вычисляемым по формуле шагом
22. Данные, содержащиеся в ячейке в электронной таблице Excel, можно редактировать
- В меню
  - В строке формул
  - В ячейке
  - В специальном окне
23. Принципиальное отличие в электронной таблице Excel круговой диаграммы от всех остальных типов диаграмм
- Может отображать только один ряд данных
  - Может отображать только одну категорию значений
  - Может показать относительную долю значений в общем результате
24. Что лучше показывает совмещенная столбиковая диаграмма в электронной таблице Excel
- Относительную долю значений в общем результате
  - Изменение суммы значений каждой категории
  - Изменение суммы значений каждой категории и относительную долю значений
25. Что лучше показывает диаграмма площадей в электронной таблице Excel
- Относительную долю значений в общем результате
  - Тенденции в изменении значений каждого ряда и относительной доли значений
  - Изменение суммы значений каждой категории
  - Изменение суммы значений каждой категории и относительную долю значений
26. Адрес ячейки в Excel – это
- Любая последовательность символов
  - Имя столбца и номер строки
  - Адрес ячейки оперативной памяти
  - Номер строки и номер столбца
27. Ячейка электронной таблицы может содержать
- Только формулу
  - Числа, формулы и текст
  - Только числа и текст
28. Число 124,56E7, записанное в ячейке таблицы, имеет следующий математический эквивалент
- 124,56000000
  - 1245600000
  - 124560000000
  - 0,000012456
29. Выберите верное обозначение строки в электронной таблице
- 18D
  - K13
  - 34

- АВ1
30. В адресе R[4]C[-2] в электронной таблице Excel
- Абсолютен номер столбца, номер строки относителен
  - Абсолютен номер строки, номер столбца относителен
  - Все адреса абсолютны
  - Адрес содержит ошибку
  - **Все адреса относительны**
31. В адресе R2C[-2] в электронной таблице Excel
- Абсолютен номер столбца, номер строки относителен
  - **Абсолютен номер строки, номер столбца относителен**
  - Все адреса абсолютны
  - Адрес содержит ошибку
  - Все адреса относительны
32. Имя ячейки используется в электронной таблице Excel в
- **Формулах**
  - **При автоматическом построении отчетов**
  - Для обозначения рядов данных в диаграммах
  - Для применения форматов чисел
33. Если в ячейке электронной таблицы отображается последовательность символов #####, то это означает, что
- Формула записана с ошибкой
  - В формуле есть ссылка на пустую клетку
  - В формуле есть циклическая ссылка
  - **Столбец недостаточно широк**
34. В каком из указанных диапазонов содержится ровно 20 клеток электронной таблицы
- **E12:F12**
  - C12:D11
  - C3:F8
  - A10:D15
35. В ячейках таблицы записаны следующие числа A1=1, A2=3, A3=5 в B1 записана формула =ЕСЛИ(A1>2;A2;A3), в B2 записана формула =ЕСЛИ(A2>2;0;1), в B3 записана формула =СУММ(B1:B2) . Тогда значение в B3 будет равно
- 6
  - 3
  - 4
  - **5**
  -
36. Какую функцию можно выбрать в электронной таблице Excel для нахождения наибольшего значения в диапазоне?
- СЧЁТ
  - ABS
  - **МАКС**
  - РОСТ
37. Какая категория функций в электронной таблице Excel содержит функцию для расчета среднего арифметического значения?
- Финансовые
  - Математические
  - **Статистические**
  - Проверка свойств и значений

38. Можно ли изменить стиль в электронной таблице Excel, в параметрах которого установлен флажок защита?
- Нет
  - **Да, но только после того, как защита будет снята**
  - Да, но при этом нельзя изменять числовые форматы
  - Да
39. Как можно скрыть отображенное примечание к ячейке в электронной таблице Excel
- Щелкнуть по примечанию правой кнопкой мыши и выполнить команду Скрыть примечание
  - **Щелкнуть по ячейке правой кнопкой мыши и выполнить команду Скрыть примечание**
  - Выделить ячейку и выполнить команду Окно/Скрыть
40. Метод конструирования алгоритма из базовых алгоритмических конструкций называется
- Конструкционным
  - Композиционным
  - **Структурным**
41. Формальное доказательство правильности алгоритма называется
- Трассировкой
  - Тестированием
  - **Верификацией**
42. При решении задачи правильно
- Упрощать максимально алгоритм за счет усложнения данных
  - Упрощать максимально данные за счет усложнения алгоритма
  - **Упрощать алгоритм за счет более адекватной структуры данных**
43. Алгоритм не обладает свойством
- Формализованности
  - Массовости
  - **Индивидуальности**
44. Существуют задачи для которых не возможно построить алгоритм
- **Да**
  - Нет
  - Не доказано
45. Экспоненциальный алгоритм может работать быстрее полиномиального
- **Да**
  - Нет
  - Не доказано
46. Существуют алгоритмы время работы которых при росте размерности задачи слабо зависит от быстродействия компьютеров
- **Да**
  - Нет
  - Не доказано
47. Существуют алгоритмы основанные на использовании датчика случайных чисел
- **Да**
  - Нет
  - Не доказано
48. Существуют задачи для которых не возможно построить приближенный алгоритм с гарантированной оценкой погрешности решения
- Да
  - Нет

- Не доказано
  - Не доказано, но скорее всего это так
49. Фрагмент вида  $y := 0; x := 12345; \text{for } i := 1 \text{ to } 3 \text{ do begin } y := y + x \bmod 10; x := \text{int}(x/10) \text{ end}$ ; вычисляет значение переменной  $y$ , равное
- 10
  - 13
  - 12
  - 15
  - 0
50. Фрагмент вида  $k := 1; s := x[1]; \text{for } i := 1 \text{ to } n \text{ do if } (s < x[i]) \text{ then begin } s := x[i]; k := i \text{ end}; \text{write}(k)$ ; выводит значение переменной  $k$ , равное:
- Максимальному элементу массива чисел  $x[1], x[2], \dots, x[n]$
  - Сумме элементов  $x[1], x[2], \dots, x[n]$ , меньших, чем  $x[1]$
  - Индексу максимального элемента ряда  $x[1], x[2], \dots, x[n]$
51. Фрагмент вида  $k := 1; s := x[1]; \text{for } i := 1 \text{ to } n \text{ do if } (s < x[i]) \text{ then begin } s := x[i]; k := i \text{ end}; \text{write}(s)$ ; выводит значение переменной  $s$ , равное:
- Максимальному элементу массива чисел  $x[1], x[2], \dots, x[n]$
  - Сумме элементов  $x[1], x[2], \dots, x[n]$ , меньших, чем  $x[1]$
  - Индексу максимального элемента ряда  $x[1], x[2], \dots, x[n]$
52. Правда ли, что
- Вызов подпрограммы занимает меньше процессорного времени, чем вызов процедуры
  - При вызове подпрограммы выделяется область памяти для локальных переменных
  - Параллельные процессы имеют общую область памяти
53. Правда ли, что
- Комментарии в исходном тексте облегчают сопровождение программы
  - Управление качеством продукта - часть системы управления организации
  - Отслеживание хода разработки проекта - проблема управления персоналом
54. Тестирование - это процесс исполнения программы
- Чтобы показать, что программа работает правильно
  - С целью выявления ошибок
  - С целью устранения ошибок
55. При тестировании проверяется
- Поведение программы при нормальных входных данных
  - Поведение программы при неправильных входных данных
  - Осмысленность сообщений об ошибках
56. Синтаксические ошибки в программе устраняют в процессе
- Отладки
  - Сопровождения
  - Тестирования