МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ

НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
“ХАРЬКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ”

Н.В. Савченко, В.Н. Кухаренко

СБОРНИК ТЕСТОВ
по курсу
“Основы информационных технологий и программирование”

Учебно-методическое пособие
для студентов физико-технических специальностей

Утверждено
Редакционно-издательским
советом университета,
протокол №3 от 15.12.2005 г.

Харьков НТУ “ХПИ” 2006

ББК 32.973.26
С 13
УДК 004.42

Рецензенты:
О.В. Серая, доц., канд. техн. наук, Национальный технический университет “Харьковский политехнический институт”;
С.А. Раков, доц., канд. физ.-мат. наук, Харьковский национальный педагогический университет им. Г.С. Сковороды

Р 19 Савченко Н.В., Кухаренко В.Н. Сборник тестов по курсу “Основы информационных технологий и программирование”: Учебно-методическое пособие – Харьков: НТУ “ХПИ”, 2006. – 75 с. На русском языке.

Пособие содержит более 400 тестов для проверки усвоения учебного материала, предусмотренного программой по дисциплине “Основы информационных технологий и программирование. Часть II.”. Для самостоятельной работы студентов физико-технических специальностей втузов.

Посібник містить набір більше 400 тестів для перевірки засвоєння учбового матеріалу, передбаченого програмою з дисципліни «Основи інформаційних технологій та програмування. Частина II.». Для самостійної роботи студентів фізико-технічних спеціальностей втузів.

Ил. 1. Табл.2. Библиогр. 24 наим.

ББК 004.42

© Савченко Н.В., Кухаренко Н.В., 2006 г.

СОДЕРЖАНИЕ

	Введение ……...……..	4
1.	Тесты по теме “Основы информатики” …….....	10
2.	Тесты по теме “Числовые типы” ……...	12
3.	Тесты по теме “Логический тип” ……..…...	14
4.	Тесты по теме “Массивы” …….....	16
5.	Тесты по теме “Подпрограммы” ……..……	22
6.	Тесты по теме “Указатели” ……..….	25
7.	Тесты по теме “Ввод-вывод” ……..……..……	29
8.	Тесты по теме “Программные модули” ……	32
9.	Тесты по теме “Обзорный тест” ……..……	35
10.	Тесты по теме “Рекурсия” ……..…...	39
11.	Тесты по теме “Сортировка и поиск” ……..……....…...	42
12.	Тесты по теме “Средства графики” ……..	45
13.	Тесты по теме “Введение в ООП” …….....	50
14.	Тесты по теме “Основы Delphi” …….....	55
15.	Тесты по теме “Язык Object Pascal” ……..	59
	Список использованной и рекомендуемой литературы	73

Кто не разбивает льда, тот не пьет холодной воды.
Ассирийская пословица [1, с. 97, п.149]

ВВЕДЕНИЕ

В настоящий сборник вошла часть тестовых заданий, предлагаемых студентам 2-го курса кафедры технической криофизики физико-технического факультета НТУ «ХПИ» при изучении курса «Основы информационных технологий и программирование. Часть II.». Главная цель этого курса состояла в том, чтобы дать студентам необходимые умения и навыки по созданию программ средней сложности на языке Паскаль. В случае успешного изучения данного курса слушатели получат необходимые практические умения и навыки для самостоятельного использования языка программирования Паскаль в своей профессиональной деятельности.

Таблица. 0.1 – Статистика базы тестов на июль 2005 года. Среднее количество карточек на тест равно 92.

№ п/п	Название теста	Количество карточек	Общий размер (байт)
1	Основы информатики	100	13254
2	Числовые типы	52	4998
3	Логический тип	30	3332
4	Массивы	110	37934
5	Подпрограммы	130	24556
6	Указатели	45	7737
7	Ввод-вывод	138	27744
8	Программные модули	51	7036
9	Обзорный тест	138	22077
10	Рекурсия	69	14121
11	Сортировка и поиск	54	12614
12	Средства графики	130	34193
13	Введение в ООП	48	9712
14	Основы Delphi	85	21237
15	Язык Object Pascal	200	42922
	Всего	1380	283467

Полный набор тестов хранится на сайте http://dl.kpi.kharkov.ua/tkf/tkf3p. Данный сайт создан с использованием виртуальной учебной среды “Веб-класс ХПИ” и выступает в качестве электронной поддержки обычных занятий. Данная технология позволяет использовать локальные и глобальные сети для одновременного параллельного тестирования всей учебной группы.

Таблица 0.2 – Трудозатраты студентов группы ФТ-23 (осенний семестр, 2004 г.) при прохождении тестирования. Максимальное количество баллов равно 275 (16 тестов). В этом пособии исключены тесты по теме «Прикладные вычисления».

№ п/п	Фамилия, Имя	Общее количество баллов	Общее количество тестирований	Число попыток на один тест	Итоговая оценка за семестр
1	Бодров Дмитрий	178	21	1	4
2	Гавриленко Виктория	154	59	4	3
3	Замлер Евгений	229	38	2	5
4	Зарицкая Анна	180	23	1	4
5	Коропец Денис	209	76	5	5
6	Крайнюков Александр	188	19	1	4
7	Ланина Наталия	201	39	2	4
8	Левченко Алексей	80	17	1	4
9	Миненко Александра	247	122	8	5
10	Пархоменко Марина	242	77	5	5
11	Первакова Алена	224	53	3	3
12	Пономаренко Павел	103	25	2	3
13	Саенко Вячеслав	226	35	2	5
14	Синица Александр	120	15	1	3
15	Сиренко Наталия	150	31	2	3
16	Смык Евгения	218	29	2	5
17	Таран Иван	57	13	1	3
18	Ушакова Елена	163	24	2	3
19	Челядинова Алина	161	37	2	3
20	Шерстюк Алексей	248	88	6	5
21	Щербина Анастасия	231	87	5	5
	Среднее значение		44	3	4

Поскольку сайт размещен на портале НТУ “ХПИ”, тестирование возможно и через глобальную сеть Интернет в любой удобный момент времени для учащегося. В данной системе преподаватель имеет возможность в режиме онлайн редактировать содержимое базы тестовых вопросов, проводить анализ результатов тестирования, составлять рейтинговые таблицы, анализировать трудность вопросов.

Для тестирования знаний используется подсистема X-тесты виртуальной учебной среды. На июль 2005 года база данных состоит из 1380 карточек-вопросов, в среднем по 90 карточек на тест. Более подробная информация о структуре базы вопросов представлена в таблице 0.1. При каждом тестировании студентам предлагается случайным образом по от 15 до 25 карточек из существующего на данный момент набора. Все ответы оцениваются одинаково: правильные – 1 балл, неправильные – 0 баллов. При прохождении тестов студентам разрешается использовать любую справочную информацию, общаться друг с другом, задавать вопросы преподавателю.

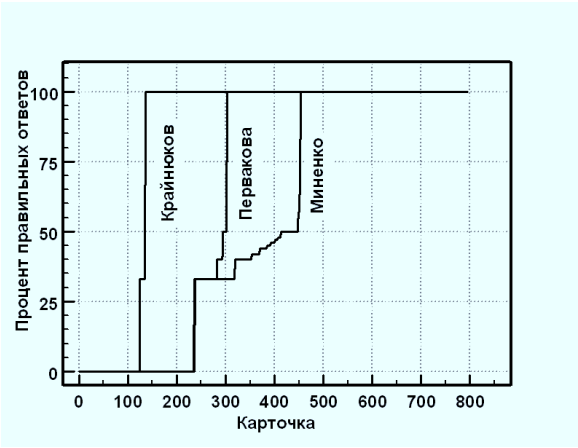


Рис.0.1. – Распределение правильных и неправильных ответов для различных студентов (Миненко – сильная; Крайнюков, Первакова – слабые) после прохождения всех тестов.

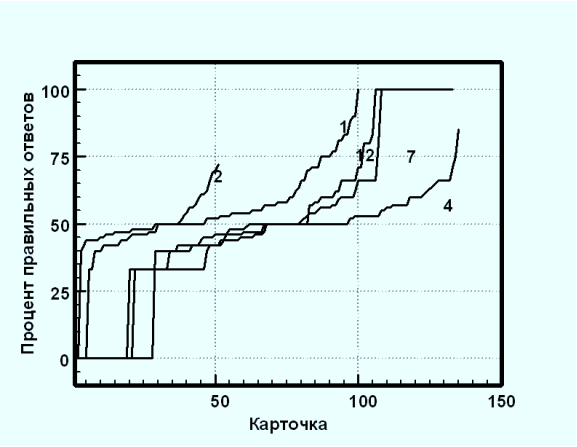


Рис.0.2. – Распределение правильных и неправильных ответов для различных тестов (1,2,4,7,12) после прохождения студентами группы ФТ-23 в осеннем семестре 2004 года.

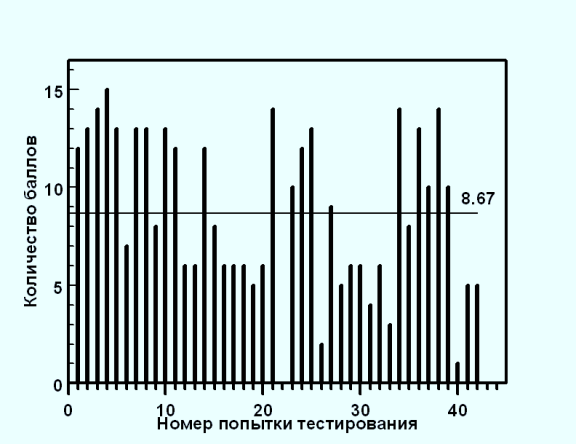


Рис.0.3. – Количество баллов, которое набирали студенты при прохождении 12-го теста. Попытки упорядочены во времени. Средний балл 8,67.

Рис.0.4. – Максимальное количество баллов, набранных студентами при прохождении 12-го теста. Среднее значение равно 10,38.

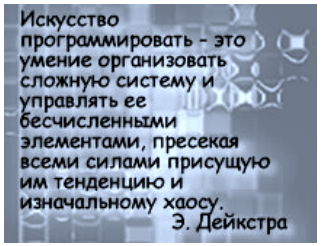
На рис. 0.1 представлены зависимости процента правильных ответов после прохождения всех тестов для сильных и слабых студентов. «Ступенчатая» зависимость кривых для слабых студентов указывает на то, что эти студенты постоянно пользовались помощью сильных студентов.

Данное пособие не содержит ответов на предлагаемые ответы. Правильность ответов студент узнает во время проведения электронного тестирования. Если студент не может найти правильный ответ, то он имеет право получить консультацию у преподавателя. Статус преподавателя позволяет в любой момент времени иметь доступ на сайте курса к базе правильных ответов.

Собранные в данной работе тестовые задания, являясь частью электронной тестовой базы, дают возможность студентам проводить предварительную самостоятельную подготовку к тестированию. Отличительной особенностью предложенного набора тестов является разнообразность форм вопросов, которые позволяют не только проверить знания, но и выработать конкретные приемы составления простых компьютерных программ.

Часть материала (книги в формате djvu, бесплатное программное обеспечение, электронный курс «Основы программирования на языке Паскаль» и др.) собраны на лазерном диске. Копия этого диска распространяется бесплатно. Каждый желающий может получить эту информацию, обратившись в Проблемную лабораторию дистанционного обучения НТУ «ХПИ», которая находится в вечернем корпусе (за электрокорпусом, 2-й этаж, левая сторона, комната 28, тел. (057) 70-76-382, интерактивный веб-сайт http://dl.kpi.kharkov.ua/techn/rle).

1. ТЕСТЫ ПО ТЕМЕ «ОСНОВЫ ИНФОРМАТИКИ».



– А в этом доме что было до исторического материализма?
– Когда было?
– Да тогда, **при старом режиме**?
– А при старом режиме барин мой жил.
– Буржуй?
– Сам ты буржуй! Он не буржуй был.
Предводитель дворянства.
– Пролетарий, значит?
– Сам ты пролетарий! Сказано тебе – предводитель.
Илья Ильф, Евгений Петров Двенадцать стульев.

1.1. Какой комбинацией клавиш в Microsoft Windows можно отметить все файлы?
□ а) *Ctrl-X*
□ б) *Ctrl-Z*
□ в) *Ctrl-C*
□ г) *Ctrl-V*
□ д) *Ctrl-V*

1.2. Какой комбинацией клавиш в Microsoft Windows можно скопировать объект в буфер?
□ а) *Ctrl-X*
□ б) *Ctrl-Z*
□ в) *Ctrl-C*
□ г) *Ctrl-V*
□ д) *Ctrl-V*

1.3. Какой комбинацией клавиш в Microsoft Windows можно вырезать объект?
□ а) *Ctrl-X*
□ б) *Ctrl-C*
□ в) *Ctrl-V*
□ г) *Ctrl-Shift*
□ д) *Ctrl-Z*

1.4. Как в Microsoft Windows можно отметить файлы выборочно?
□ а) *Ctrl-A*
□ б) *Ctrl-Z*
□ в) *Ctrl+щелчок мыши*
□ г) *Alt+щелчок мыши*
□ д) *Ctrl-Alt-Del*

1.5. Укажите фирмы, которые не производят микропроцессоры:
□ а) *Intel;*
□ б) *Oriflame;*
□ в) *AMD;*
□ г) *IBM;*
□ д) *McDonalds;*

1.6. Какие характеристики не являются существенными для жестких дисков:
□ а) емкость;
□ б) вес;
□ в) быстродействие;
□ г) интерфейс;

□ е) *Cyrix;*

□ ж) *Motorolla.*

1.7. Человек, пишущий команды для введения их в компьютер, называется:

□ а) кодировщиком;

□ б) халтурщиком;

□ в) стукачом;

□ г) программистом;

□ д) печатником.

1.9. Преобразовать в двоичный код следующее десятичное число: $3910_{(10)}$ = _____.

1.11. Преобразовать в шестнадцатеричный код следующее двоичное число: $11002_{(2)}$ =_____.

1.13. 1 Кб =_____ байт.

1.15. О_____ система - набор системных программ, управляющих работой компьютерной системы. Она также поддерживает взаимодействие между частями компьютерной системы.

1.17. Основная характеристика жесткого диска - это его _____.

1.19. Байт содержит_____ (цифра) бит и представляет числовое значение от 0 до ____ (десятичное).

□ д) фирма изготовитель.

1.8. В лазерном принтере используется принцип:

□ а) монографии;

□ б) ксерографии;

□ в) полиграфии;

□ г) томографии;

□ д) каллиграфии.

1.10. Преобразовать в десятичный код следующее двоичное число: $10000000_{(2)}$ = _____.

1.12. Преобразовать в десятичный код следующее шестнадцатеричное число: $34CF_{(16)}$ = _____.

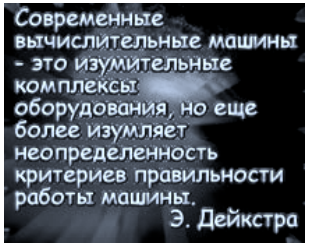
1.14. 64 Кб = 6___536 байт.

1.16. _____ память - память, размещенная в компьютере, используется для хранения программы и ее данных во время ее выполнения.

1.18. Все основные узлы компьютера располагаются в _____ блоке.

1.20. _____ - это наименьшая единица хранимой информации.

2. ТЕСТЫ ПО ТЕМЕ «ЧИСЛОВЫЕ ТИПЫ»



– Вот! **Вот наше будущее, настоящее и прошедшее!** Зажигайте, Киса, спички, а я его вскрою.

И Остап полез в карман за инструментами.– Ну-с, – сказал он, протягивая руку к стулу, – еще одну спичку, предводитель. Вспыхнула спичка, и, странное дело, стул сам собой скакнул в сторону и вдруг, на глазах изумленных концессионеров, провалился сквозь пол.

Илья Ильф, Евгений Петров Двенадцать стульев.

2.1. Указать неправильные записи чисел в Паскале:

□ а) 7.

□ б) -0

□ в) 7,0

□ г) 0006

2.3. Записать на Паскале следующую формулу (a,b,c,x,y,z - переменные): a+bx+cyz

Ответ: _____.

2.5. Записать на Паскале следующее число 5!

Ответ: _____.

2.7. Значение выражения: *20 mod 4* равно _____.

2.9. Записать следующие числа без десятичного порядка: 666E-3

Ответ: _____.

2.11. Вычислить значение выражения *trunc(6.9).*

Ответ: _____.

2.13. Тип выражения sqrt(16):

□ а) целый;

□ б) вещественный.

2.2. Указать неправильные записи чисел в Паскале:

□ а) 0E-4

□ б) 2*E5

□ в) 8E0

□ г) e

2.4. Сколько операций выполняется при вычислении выражения: (x+1/2)*(y+7/10)-3/4

Ответ: _____.

2.6. Записать на Паскале следующее число 1/100000.

Ответ: _____.

2.8. Значение выражения: *20 div 6* равно _____.

2.10. Вычислить значение выражения: *3.0 mod 3*

Ответ: _____.

2.12. Вычислить значение выражения *round(6.9).*

Ответ: _____.

2.14. Тип выражения sin(0):

□ а) целый;

□ б) вещественный.

□ в) логический

2.15. Тип выражения odd(3):

□ а) целый;

□ б) вещественный.

□ в) логический

2.17. Вычислить значение выражения: 123 div 0

Ответ: _____.

2.19. Записать на Паскале следующее число LXIV

Ответ: _____.

2.21. Переменная типа *Shortint* (короткое целое) занимает в памяти компьютера

□ а) 1 байт

□ б) 2 байта

□ в) 3 байта

□ г) 4 байта

□ д) 5 байт

□ е) 6 байт

2.23. Переменная типа *Real* занимает в памяти компьютера

□ а) 2 байт

□ б) 4 байта

□ в) 6 байта

□ г) 8 байта

□ д) 10 байт

□ е) 12 байт

□ в) логический

2.16. Записать на Паскале следующую формулу: x^4

Ответ: _____.

2.18. Какое значение будет иметь переменная x после выполнения операторов x:=10; x:=x+3;?

Ответ: x=_____.

2.20. Вычислить:

24/(3*4)-24/3/4+24/3*4

Ответ: _____.

2.22. Переменная типа *Longint* (длинное целое) занимает в памяти компьютера

□ а) 1 байт

□ б) 2 байта

□ в) 3 байта

□ г) 4 байта

□ д) 5 байт

□ е) 6 байт

2.24. Переменная типа *Double* занимает в памяти компьютера

□ а) 2 байт

□ б) 4 байта

□ в) 6 байта

□ г) 8 байта

□ д) 10 байт

□ е) 12 байт

3. ТЕСТЫ ПО ТЕМЕ «ЛОГИЧЕСКИЙ ТИП»

Великий **комбинатор был прав**, как всегда. Ипполит Матвеевич оправился от смущения только в ту минуту, когда с палубы понеслись звуки увертюры, исполняемой на кружках Эсмарха и пивных батареях.

Илья Ильф, Евгений Петров Двенадцать стульев.

3.1. Вычислить значение выражения:

sqr(x)+sqr(y)<=4, при x=0.3, y=-1.6

Ответ:

□ а) *true*

□ б) *false*

3.3. Записать на Паскале отношение, истинное при выполнении указанного условия и ложное в противном случае:

целое k делиться на число 7.

Ответ: _____.

3.5. Вычислить значение выражения:

not odd(n), при n=0

Ответ:

□ а) *true*

□ б) *false*

3.7. Записать на Паскале отношение, истинное при выполнении указанного условия и ложное в противном случае:

3.2. Вычислить значение выражения:

k mod 7 = k div 5 -1 , при k=15

Ответ:

□ а) *true*

□ б) *false*

3.4. Записать на Паскале отношение, истинное при выполнении указанного условия и ложное в противном случае:

точка (x,y) лежит вне круга радиуса r с центром в точке (1,0). При записи использовать только функцию sqr.

Ответ: _____.

3.6. Вычислить следующее выражение при a=true и b=false:

a or b and not a

Ответ:

□ а) *true*

□ б) *false*

3.8. Записать на Паскале отношение, истинное при выполнении указанного условия и ложное в противном случае:

уравнение $ax^2 + bx + c = 0\ (a \neq 0)$ не имеет вещественных корней.
Ответ: _____.

3.9. Вычислить значение выражения:
t and (p mod 3 = 0), при t=true, p=101010
Ответ:
□ а) *true*
□ б) *false*

3.11. Вычислить:
a or a, при a= true или false
Ответ:
□ а) *a*
□ б) *true*
□ в) *false*

3.13. Вычислить значение выражения:
ord(false)=1
Ответ:
□ а) *true*
□ б) *false*

3.15. Вычислить значение выражения:
a or (not b), при a=false, b=true
Ответ:
□ а) *true*
□ б) *false*

3.17. Вычислить:
a and (not a), при a= true или false
Ответ:
□ а) *a*
□ б) *true*
□ в) *false*

натуральное число n является полным квадратом.
Ответ: _____.

3.10. Вычислить значение выражения:
(x*y<>0) and (y>x), при x=2, y=1
Ответ:
□ а) *true*
□ б) *false*

3.12. Вычислить:
false and a, при a= true или false
Ответ:
□ а) *a*
□ б) *true*
□ в) *false*

3.14. Вычислить значение выражения:
pred(true)
Ответ:
□ а) *true*
□ б) *false*

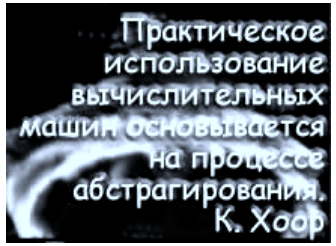
3.16. Вычислить значение выражения:
*(x*y<>0) or (y>x)*, при *x=2, y=1*
Ответ:
□ а) *true*
□ б) *false*

3.18. Вычислить:
a or (not a), при *a= true* или *false*
Ответ:
□ а) *a*
□ б) *true*
□ в) *false*

3.19. Вычислить:
true or (1/0>0)
Ответ:
□ а) *true*
□ б) *false*
□ в) *error*

3.20. Вычислить:
(1/0>0) or true
Ответ:
□ а) *true*
□ б) *false*
□ в) *error*

4. ТЕСТЫ ПО ТЕМЕ «МАССИВЫ»



Дверные приборы были страстью Александра Яковлевича. Положив великие труды, он снабдил все без исключения двери пружинами **самых разнообразных систем и фасонов**. Здесь были простейшие пружины, в виде железной штанги. Были духовые пружины с медными цилиндрическими насосами. Были приборы на блоках со спускающимися увесистыми дробовыми мешочками. Были еще пружины конструкций таких сложных, что собесовский слесарь только удивленно качал головой. Все эти цилиндры, пружины и противовесы обладали могучей силой.
Илья Ильф, Евгений Петров Двенадцать стульев.

4.1. В качестве имени массива может быть использован любой
□ а) пластификатор;
□ б) мистификатор;
□ в) модификатор;
□ г) скарификатор;
□ д) идентификатор;
□ е) версификатор.

4.2. Обращение к элементам массива осуществляется с помощью указания имени массива и списка индексов
□ а) между наклонными чертами;
□ б) в круглых скобках;
□ в) в фигурных скобках;
□ г) между апострофами;
□ д) в квадратных скобках;
□ е) между знаками меньше и больше.

4.3. В качестве разделителя в списке индексов многомерного массива используется

4.4. Какие из перечисленных ниже записей не являются обращениями к элементам массива

□ а) знак доллара;
□ б) наклонная черта;
□ в) точка с запятой;
□ г) запятая;
□ д) точка;
□ е) двоеточие.

4.5. Какие из перечисленных ниже служебных слов не могут быть использованы при объявлении массива
□ а) *array*
□ б) *of*
□ в) *integer*
□ г) *const*
□ д) *type*
□ е) *var*

4.7. Какие из перечисленных ниже записей не являются обращениями к элементам массива
□ а) *MAMA[1,1]*
□ б) *ПАЛЬТО[1,1]*
□ в) *КЕНГУРУ[1,1]*
□ г) *ВАРВАРА[1,1]*
□ д) *DOG'S[1,1]*

4.9. Индексы в массиве могут быть любого порядкового типа, кроме типа
□ а) *Integer*
□ б) *Char*
□ в) *LongInt*

4.11. Следующий фрагмент кода некоторой программы

□ а) A[1,1]
□ б) A[1:1]
□ в) A[1;1]
□ г) A[1-1]
□ д) A[1,1,1]

4.6. Какие из перечисленных ниже служебных слов не могут быть использованы при объявлении массива
□ а) *function*
□ б) *end*
□ в) *begin*
□ г) *for*
□ д) *while*
□ е) *of*

4.8. Какие из перечисленных ниже записей не являются обращениями к элементам массива
□ а) *MAS[1,sqr(4)]*
□ б) *MAS[1,sqrt(5)]*
□ в) *MAS[1,sqrt(4)]*
□ г) *MAS[sqr(4),sqr(2)]*
□ д) *MAS[1/10,10/1]*
□ е) *MAS[1+6,sqr(5)+1]*

4.10. Для двумерного массива считают, что первый индекс обозначает номер _____, а второй - номер _____ воображаемой таблицы, элементами которой являются соответствующие значения данного массива.

4.12. Следующий фрагмент кода некоторой программы

Const
MaxN=10;
MaxM=15;
var
tmyarray :
array[1..MaxN,1..MaxM] of integer;
определяет массив с именем _____.

4.13. В программе необходимо будет работать с массивами такого типа:

false	false	true
false	true	false

Допишите недостающую информацию в фрагмент кода программы, решающий эту задачу
Const
MaxN= _____ ;
Var
MaxM= _____;
tmyarray : array[1..MaxN,1..MaxM]
of _____ ;

4.15. Количество элементов в двумерном массиве определяется путем умножения числа строк на число _____ этого массива.

4.17. В программе определен двумерный массив *tmyarray*.

Const
MaxN=3;
MaxM=3;
tmyarray : array[1..MaxN,1..MaxM]
of integer = ((1,0,0),(0,1,0),(0,0,1));
определяет массив-константу. Постройте соответствующую ей воображаемую матрицу. Ответ:

___	___	___
___	___	___
___	___	___

4.14. В программе определен двумерный массив *tmyarray*.
Const
MaxN= 2;
MaxM= 3;
Var
tmyarray : array[1..MaxN,1..MaxM]
of boolean;

Количество строк в этом массиве равно _____.

4.16. Количество необходимой памяти для хранения массива определяется путем умножения числа элементов массива на число _____, требуемое для хранения отдельной компоненты массива.

4.18. В программе определен двумерный массив *tmyarray*.

```
Const
MaxN= 2;
MaxM= 3;
Var
tmyarray :
array[1..MaxN,1..MaxM] of boolean;
```
Количество элементов в этом массиве равно _____.

4.19. Формула для определения адреса ячейки памяти, где хранится i,j элемент двумерного массива *Mas* имеет вид
addr(Mas[i,j]] = addr(Mas[1,1])+V(M*(i-1)+j-1)*, где

M - количество _____;
V - число байт, необходимых для хранения элемента массива.

4.21. Если
```
Var a,b: array [1..5] of integer;
c: array[1..5] of integer;
```
то оператор *b:=a+b*;

□ а) допустим
□ б) не допустим

4.23. Какие значения будут храниться в массиве B после выполнения следующих операторов
```
var
b:array[1..3,1..3] of integer;
i,j: integer;
begin
for i:=1 to 3 do
for j:=1 to 3 do
```
```
Const
MaxN= 2;
MaxM= 3;
Var
tmyarray :
array[1..MaxN,1..MaxM] of boolean;
```
Объем памяти, необходимый для хранения этого массива, равен _____ байт.

4.20. Для организации доступа ко всем элементам двумерного массива необходимо использовать вложенные _____.

4.22. Элементы, оба индекса которых равны, образуют главную _____ матрицы.

4.24. Какие значения будут храниться в массиве B после выполнения следующих операторов
```
var
B:array[1..3,1..3] of integer;
i,j: integer;
begin
for i:=1 to 3 do
for j:=1 to 3 do
```



```
begin
b[i,j]:=0
if i=j then b[i,j]:=1
end;
```
Ответ:

____	____	____
____	____	____
____	____	____

4.25. Известно, что в точке T1 массив *a* содержал указанные значения. Определите содержимое массива *a* в точке T2 в результате работы фрагмента алгоритма.
Массив *a* в {T1} Фрагмент программы Массив *a* в {T2}

11	12	13
0	4	3
-1	0	14

{T1}
```
for i:=1 to 3 do
for j:=1 to i do
begin
c=a[i,j];
a[i,j]:=a[j,i];
a[j,i]:=c
end;
```
{T2}
Ответ:

____	____	____
____	____	____
____	____	____

4.27. Задан двумерный массив *a[1..n,1..n]*. Фрагмент алгоритма


```
b[i,j]:=i-j;
```
Ответ:

____	____	____
____	____	____
____	____	____

4.26. Известно, что в точке T1 массив *a* содержал указанные значения. Определите содержимое массива *a* в точке T2 в результате работы фрагмента алгоритма.
Массив *a* в {T1} Фрагмент программы Массив *a* в {T2}

5	6	7
8	9	10
1	2	3

{T1}
```
n:=3;
for i:=1 to (n div 2) do
for j:=1 to n do
begin
c=a[i,j];
a[i,j]:=a[n-i+1,j];
a[n-i+1,j]:=c
end;
```
{T2}
Ответ:

____	____	____
____	____	____
____	____	____

4.28. Задан двумерный массив *a[1..n,1..n]*. Фрагмент алгоритма

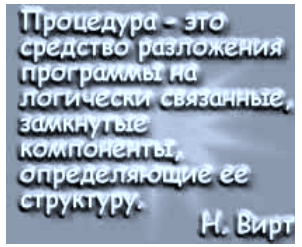
```
for i:=1 to n do
begin
c:=a[i,i];
a[i,i]:=a[i,n-i+1];
a[i,n-i+1]:=c
end;
```

□ а) меняет порядок строк таблицы
□ б) меняет порядок столбцов
□ в) меняет местами элементы главной и побочной диагонали
□ г) меняет местами элементы i-ой строки и побочной диагонали
□ д) ничего не меняет в таблице

```
for i:=1 to n do
for j:=1 to n do
begin
c:=a[i,j];
a[i,j]:=a[n-i+1,j];
a[n-i+1,j]:=c
end;
```

□ а) меняет порядок строк таблицы
□ б) меняет порядок столбцов
□ в) меняет местами элементы главной и побочной диагонали
□ г) меняет местами элементы i-ой строки и побочной диагонали
□ д) ничего не меняет в таблице

5. ТЕСТЫ ПО ТЕМЕ «ПОДПРОГРАММЫ»



– Следующим номером нашей концертной программы выступит мировая исполнительница русских народных песен, хорошо **известная в Марьиной Роще, Варвара Ивановна Годлевская**.
Варвара Ивановна!
Пожалуйте! Ипполит Матвеевич пил водку и молчал. Так как Лиза не пила и все время порывалась уйти домой, надо было спешить, чтобы успеть выпить весь графин.
Илья Ильф, Евгений Петров Двенадцать стульев.

5.1. _____ - это повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы.

5.2. В языке Паскаль имеется две разновидности подпрограмм - процедуры и _____.

5.3. Программы на языке Паскаль состоят из:
заголовка программы,
раздела _____ и
тела программы.

5.4. Раздел описаний может включать следующие подразделы объявлений:
меток,
констант,
_____,
переменных,
процедур и
функций.

5.5. Последовательность подразделов в структуре программы произвольная, но обязательно должно выполняться правило _____ описания.

5.6. Все параметры (константы, переменные, типы) в программе делятся на _____ и локальные.

5.7. Глобальные параметры доступны как в основной _____ так и во всех подпрограммах.

5.8. Число и _____ формальных и фактических параметров подпрограмм должны совпадать с точностью до их

5.9. При описании подпрограмм формальные параметры указываются в виде списка имен с типом через
□ а) запятую;
□ б) точку с запятой;
□ в) двоеточие;
□ г) точку;
□ д) тире.

5.11. Параметры-переменные - это параметры, передаваемые по _____.

5.13. В списке формальных параметров подпрограммы перед параметром-переменной должен стоять идентификатор
□ а) *Const*
□ б) *Var*
□ в) *Type*
□ г) *Uses*

5.15. Правило структурного программирования относительно использования подпрограмм: Каждая процедура должна иметь одну точку входа и одну точку _____.

5.17. В отличие от основной программы процедура завершается не точкой, а

следования.

5.10. При описании подпрограмм формальные параметры указываются в виде списка имен с типом через точку с запятой и помещаются в
□ а) фигурные
□ б) квадратные
□ в) круглые
□ г) угловые
скобки.

5.12. Параметры-_____ - это когда копия фактического параметра становится значением соответствующего формального параметра.

5.14. Действие ключевого слова *Var* в списке формальных параметров подпрограммы распространяется до ближайшей
□ а) точки;
□ б) двоеточия;
□ в) точки с запятой;
□ г) тире;
□ д) круглой скобки.

5.16. Заголовок процедуры состоит из зарезервированного слова, за которым следует _____ имени процедуры, а далее в круглых скобках список формальных _____.

5.18. Процедура ввода N-чисел. Введите недостающий текст
Пусть в основной программе определен тип:

□ а) восклицательным знаком;
□ б) двоеточием;
□ в) точкой с запятой;
□ г) троеточием;
□ д) знаком доллара;
□ е) вопросительным знаком.

5.19. Заголовок функции в Паскале начинается с ключевого слова
□ а) *main*
□ б) *function*
□ в) *procedure*
□ г) *subroutine*

5.21. Заголовок функции имеет вид:
function TestDiag:boolean;
В этой функции количество формальных параметров равно
□ а) 0
□ б) 1
□ в) 2
□ г) 3

Type tarr: array[1..100] of integer;
Procedure InpInt(var Mas: tarr; N: integer);
Var i: integer;
Begin
Write('Введите ', N, ' чисел');
For i:=1 to N do
Read(_____)
End;

5.20. Заголовок функции имеет вид:
function Intersection(ax1, ay1, ax2, ay2, bx1, by1, bx2, by2:real):boolean;
В этой функции тип формальных параметров -
□ а) целый;
□ б) вещественный;
□ в) логический;
□ г) строковый;
□ д) символьный.

5.22. Дана функция:
function TestDiag:boolean;
var j,s:integer;
begin
s:=0;
*for j:=1 to N do s:=s+a[(N-j)*N+j];*
TestDiag:=(s=IdealSum);
end;
Здесь переменные *j* и *s* являются
□ а) глобальными
□ б) локальными
□ в) формальными
□ г) фактическими
параметрами.

5.23. Дана функция:

```
function TestDiag:boolean;
var j,s:integer;
begin
s:=0;
for j:=1 to N do s:=s+a[(N-j)*N+j];
TestDiag:=(s=IdealSum);
end;
```

Здесь *N* является

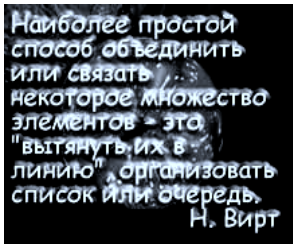
□ а) глобальным
□ б) локальным
□ в) формальным
□ г) фактическим

параметром.

5.24. Для досрочного выхода из подпрограммы может быть использована функция

□ а) *End*
□ б) *Quit*
□ в) *Exit*
□ г) *Stop*
□ д) *Return*

6. ТЕСТЫ ПО ТЕМЕ «УКАЗАТЕЛИ»



– Что вы, Бендер, смотрите, **какая очередь стоит**! Неминуемо побьют!– Об этом не думайте. Когда будут бить – будете плакать, а пока что не задерживайтесь! Учитесь торговать!

Илья Ильф, Евгений Петров Двенадцать стульев.

6.1. Write(LST, #12, #13, #10) то же, что и:

□ а) Write(LST, ^L, ^M, ^J);
□ б) Writeln(LST, ^L);
□ в) Writeln(LST, #12);
□ г) Все из них

6.2. Mem[Seg(P^):Ofs(P^)] = P^

□ а) Верно если SizeOf(P^) = 1;
□ б) Верно если SizeOf(P^) > 1;
□ в) Всегда неверно;
□ г) Всегда верно.

6.3. Что будет выдано на экране

6.4. Что будет выдано на экране в результате выполнения следующих операторов?

```
program prlink;
type ref = ^integer;var   p,q: ref;
begin
new(p); p^:=5; new(q); q^:=3;
p^:=q^;
if p=q then p:=nil else if p^=q^ then
q:=p;
if p=q then q^:=4;write(p^)end.
```

Ответ: _____.

6.5. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
p:=q;
```

□ а) Да
□ б) Нет

6.7. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
p:=nil;
```

□ а) Да
□ б) Нет

6.9. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
q:=p^;
```

□ а) Да
□ б) Нет

6.4. Что будет выдано на экране в результате выполнения следующих операторов?

```
program prlink;
type d = record a: boolean; b,c:^real;
end;
var   r: ^d;
begin
new(r); r^.a:=false;
r^.c:=nil;new(r^.b); r^.b^:=2.7;
if r^.b<>nil then r^.c:=r^.b;
r^.b^:=r^.c^-1.4; r^.a:=r^.b=r^.c;
write(r^.a,r^.b^:4:1)end.
```

Ответ: _____.

6.6. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
q:=r;
```

□ а) Да
□ б) Нет

6.8. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
r:=nil;
```

□ а) Да
□ б) Нет

6.10. Пусть

```
var   p,q: ^integer;   r: ^char;
```

Следующий оператор неправильный?

```
p^:=nil;
```

□ а) Да
□ б) Нет

6.11. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
if r<>nil then r^:=nil^;
□ а) Да
□ б) Нет

6.13. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
q^:=ord(r^);
□ а) Да
□ б) Нет

6.15. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
if q=nil then write(q);
□ а) Да
□ б) Нет

6.17. New(P) :

□ а) Выделяет 8 байт
□ б) Выделяет SizeOf(P^) байт
□ в) Выделяет 1 байт
□ г) Выделяет 8 байт или SizeOf(P^) байт

6.19. Для объявления типизированного указателя перед базовым типом ставится знак указателя
□ а) ^

6.12. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
r^:=p^;
□ а) Да
□ б) Нет

6.14. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
if q>nil then q^:=p^;
□ а) Да
□ б) Нет

6.16. Пусть
var p,q: ^integer; r: ^char;
Следующий оператор неправильный?
if q<>r then read(r^);
□ а) Да
□ б) Нет

6.18. Указатель называется неопределенным если его значение равно
□ а) *0*
□ б) *nil*
□ в) *empty*
□ г) *null*
□ д) *disannul*

6.20. В Паскале обычно указатель связывается с некоторым типом данных. Этот тип данных называется базовым, а указатель
□ а) монотипным;

□ б) @
□ в) $
□ г) &
□ д) %

6.21. В Паскале работа с динамической памятью позволяет эффективно использовать оперативную память компьютера. Такая возможность связана с наличием типа данных:
□ а) списки;
□ б) очереди;
□ в) стеки;
□ г) отсылки;
□ д) стрелки;
□ е) персты;
□ ж) указатели;
□ з) сажени.

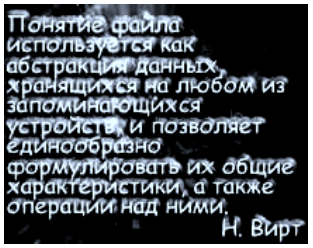
6.23. В Паскале функция, которая выделяет динамическую область памяти для размещения переменной и возвращает в типизированный указатель адрес первого байта этой памяти, имеет имя
□ а) *Dispose*
□ б) *GetMem*
□ в) *AllocMem*
□ г) *SizeOf*
□ д) *New*
□ е) *FreeMem*
□ ж) *Addr*
□ з) *Ptr*

□ б) автотипным;
□ в) типизированным;
□ г) стереотипным;
□ д) типическим.

6.22. В Паскале для того чтобы получить значение, с которым связан указатель, необходимо после имени указателя поставить знак
□ а) @
□ б) &
□ в) ^
□ г) %
□ д) *

6.24. В Паскале функция, которая освобождает динамическую область памяти, выделенную под типизированный указатель, имеет имя
□ а) *Dispose*
□ б) *GetMem*
□ в) *AllocMem*
□ г) *SizeOf*
□ д) *New*
□ е) *FreeMem*
□ ж) *Addr*
□ з) *Ptr*

7. ТЕСТЫ ПО ТЕМЕ «ВВОД-ВЫВОД»



– Что вы делаете? – спросил управдом. – Где это видано? Как можно **вводить** лошадь в жилую квартиру?
Шаринов вдруг рассердился.
– Какое тебе дело? Купил лошадь. Где поставить? Во дворе украдут!
... Как же мы будем жить, когда рядом лошадь?
Мы **пишем оперу**, нам это неудобно! - завопил Ляпис.
Илья Ильф, Евгений Петров Двенадцать стульев.

7.1. _____ – это именованная область памяти компьютера (жесткого диска, гибкой дискеты, электронного виртуального диска и т.д.), потенциальный источник и приемник информации.

7.2. В языке Паскаль ввод-вывод информации осуществляется через _____ переменные.

7.3. В Паскале имеется три категории файлов: _____ (тип *text*), типизированные (*file of* <тип>) и файл без типа (*file*).

7.4. Обычно все файлы в Паскале считаются файлами _____ доступа. Однако для типизированных файлов и файлов без типа с помощью процедуры *seek* можно установить режим произвольного доступа. При этом нумерация компонент файла начинается с нуля.

7.5. Для связи файловой переменной с внешним файлом в Паскале используется процедура
- □ а) *close(f)*
- □ б) *rewrite(f)*
- □ в) *erase(f)*
- □ г) *reset(f)*
- □ д) *assign(f,name)*
- □ е) *rename(f,name)*

7.6. Для открытия нового файла используется процедура
- □ а) *close(f)*
- □ б) *rewrite(f)*
- □ в) *erase(f)*
- □ г) *reset(f)*
- □ д) *assign(f,name)*
- □ е) *rename(f,name)*

7.7. Функция _____(f) принимает значение *true*, если указатель текущей компоненты файла находится за последней компонентой файла (за последним символом, если файл тестовый), *false* - противном случае.

7.8. Файл типа *Text* может быть открыт либо для чтения процедурой *Reset*, либо для записи процедурой *Rewrite* или _____.

7.9. Для текстового файла в Паскале процедура для чтения информации из файла имеет имя _____.

7.10. Функция *seekeof(f)* отличается от функции *eof(f)* тем, что стоящие в конце файла символы пробела и _____ пропускаются.

7.11. Оператор для ввода информации из текстового файла с именем F имеет вид:
read(_____ ,<список ввода>);
или
readln(_____,<список ввода>);

7.12. Функция *Seek(F,Num)* - настройка на требуемую компоненту файла. Осуществляется настройка на компоненту файла, с которым связана файловая переменная *f*. Компонента файла определяется номером *num*, причем нумерация компонент начинается с _____.

7.13. Функция _____(*F*) определяет номер текущей компоненты файла. Функция возвращает номер текущей компоненты файла, с которым связана файловая переменная *F*. Нумерация компонент начинается с нуля.

7.14. В Паскале логическое имя принтера
- □ а) *con*
- □ б) *prn*
- □ в) *aux*
- □ г) *nul*
- □ д) *lpt1*
- □ е) *com1*

7.15. Имя диска - это один из символов A..Z, после которого ставится
- □ а) слэш

7.16. В некоторой программе описаны файлы
```
type
product = record
name: string; code:word; cost:comp
```

□ б) двоеточие
□ в) точка с запятой
□ г) тире
□ д) знак доллара
□ е) знак вопроса

7.17. При работе с файлами автоматически проверяются ошибки ввода-вывода. Если проверка ошибки включена (по умолчанию или с помощью ключа компилятора *{$I+}*), то при возникновении ошибки выполнение программы автоматически завершается, если же проверка ошибки отключена (с помощью ключа *{$I-}*), программа продолжает выполняться и можно проверить результат ввода-вывода с помощью функции _____.

7.19. При вводе информации с клавиатуры символ конца файла *EOF(#26)* вводится при одновременном нажатии на клавиши
□ а) *Ctrl+F*
□ б) *Ctrl+E*
□ в) *Ctrl+Z*
□ г) *Alt+Z*

```
end;
text80 = file of string[80];
var
f1: file of char;
f2: text;
f3: file;
f4: text80;
f5: file of product;
```

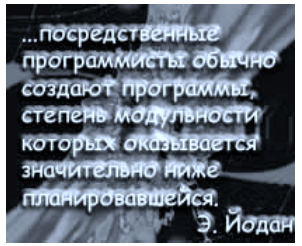
Здесь файл f2 является
□ а) текстовым
□ б) нетипизированным
□ в) типизированным

7.18. Функция *FileSize(F)* определяет текущий _____ файла. Функция возвращает текущий размер файла, с которым связана файловая переменная f, в компонентах этого файла.

7.20. Длина любого компонента типизированного файла строго постоянна, что дает возможность организовать _____ доступ к каждому из них (т.е. доступ к компоненту по его порядковому номеру).

□ д) *Alt+F*
□ е) *Alt+E*

8. ТЕСТЫ ПО ТЕМЕ «ПРОГРАММНЫЕ МОДУЛИ»



Изнуренков умудрялся острить в тех областях, где, казалось, уже ничего смешного нельзя было сказать. Из такой чахлой пустыни, как вздутые накидки на себестоимость, Изнуренков умудрялся выжать около сотни шедевров юмора... Через десять минут тема была обработана, обдуман рисунок и приделан **заголовок**.
Илья Ильф, Евгений Петров Двенадцать стульев.

8.1. Исходный текст программного модуля хранится в файле с расширением
□ а) *tpl*
□ б) *tpu*
□ в) *pas*
□ г) *txt*
□ д) *exe*
□ е) *com*

8.2. Откомпилированный программный модуль хранится в файле с расширением
□ а) *tpl*
□ б) *tpu*
□ в) *pas*
□ г) *txt*
□ д) *exe*
□ е) *com*

8.3. Каждый программный модуль транслируется _____.

8.4. Модуль содержит описание многих _____ и функций.

8.5. Программный модуль подключается к программе с помощью оператора
□ а) *make*
□ б) *with*
□ в) *include*
□ г) *uses*
□ д) *use*

8.6. Структура программного модуля

```
Unit имя_модуля;
Interface { _____ раздел}
Implementation {раздел реализации}
Begin {раздел инициализации}
end.
```

8.7. При подключении модулей в

8.8. Процедуры и функции для

программе используется оператор:
uses <список модулей>
Элементами списка являются имена модулей отделяемые друг от друга

□ а) точкой с запятой
□ б) точкой
□ в) запятой
□ г) тире
□ д) двоеточием
□ е) апострофом

8.9. В разделе _____ находится полное описание всех процедур и функций, а также описание констант, типов и переменных, которые нужны для внутреннего использования и не должны быть видимы снаружи.

8.11. Раздел _____ содержит те операторы, что будут выполнены до начала программы, к которой подключен модуль.

8.13. Стандартный модуль, который делает доступным вывод текста на принтер, называется _____.

8.15. Стандартные модули system, crt и dos физически размещены в файле _____.tpl

8.17. Имя программного модуля должно совпадать с именем дискового _____, в который помещается исходный текст модуля.

работы со строками хранятся в стандартном модуле

□ а) *dos*
□ б) *graph*
□ в) *system*
□ г) *overlay*
□ д) *crt*
□ е) *printer*

8.10. В разделе реализации находится полное описание всех процедур и функций, а также описание констант, типов и переменных, которые нужны для внутреннего использования и не должны быть _____ снаружи.

8.12. В _____ отделе перечисляется все, что должно быть видимым в модуле из программы, которая его использует.

8.14. Стандартный модуль, который обеспечивает управление текстовым режимом экрана и звуком, называется _____.

8.16. Последняя часть модуля - секция инициализации может быть пустой и содержать только зарезервированное слово _____.

8.18. На имя модуля операционная система не накладывает определенные ограничения. Это утверждение

8.19. Системный модуль, который подключается к любой программе автоматически, называется _____.

истинно.

□ а) Да
□ б) Нет

8.20. Не запрещается явное или косвенное обращение модуля к самому себе. Это утверждение истинно.

□ а) Да
□ б) Нет

9. ТЕСТЫ ПО ТЕМЕ «ОБЗОРНЫЙ ТЕСТ»

Стулья были свежие и чистые, как будто бы только что вышли из-под надзора рачительной Клавдии Ивановны.–
Они? – спросил Остап.
– Боже, Боже, – твердил Ипполит Матвеевич, – они, они.
Они самые. На этот раз сомнений никаких.
– **На всякий случай проверим**, – сказал Остап, стараясь быть спокойным.
Илья Ильф, Евгений Петров Двенадцать стульев.

9.1. Какие функции возвращают результат типа char?
- □ а) *Pred, Succ, Chr*
- □ б) *Pred, Succ, Sqrt*
- □ в) *Pred, Succ, Ord*
- □ г) Все из них

9.2. Переменные каких типов нельзя использовать в операторе case?
- □ а) *boolean*
- □ б) *char*
- □ в) *integer*
- □ г) *string*

9.3. Какое утверждение о цикле *for* верно?
- □ а) Цикл *for* выполняется по крайней мере один раз
- □ б) Индексная переменная всегда увеличивается на 1
- □ в) Индексная переменная не может модифицироваться внутри цикла
- □ г) Цикл *for* может быть неопределенным

9.4. Оператор
if 0 = true then ClrScr;
означает
- □ а) Экран будет очищен
- □ б) Экран не будет очищен
- □ в) Выражение не допустимо в Паскале

9.5. Переменные какого типа не могут быть индексными переменными в цикле?
- □ а) *integer*
- □ б) *byte*
- □ в) *real*
- □ г) *char*

9.6. Оператор
for i := $01 to $100 do;
- □ а) Будет выполнен 100 раз
- □ б) Будет выполнен 256 раз
- □ в) Этот оператор недопустим в Паскале

9.7. Оператор
while true do; означает
- □ а) Бесконечный цикл
- □ б) Выполнится один раз и завершится
- □ в) Выполняться не будет
- □ г) Выражение не допустимо в Паскале

9.8. Какие из циклов не будут выполнены?
- □ а) *while (2 > 3) or (5 < 10) do*
- □ б) *while true and false do*
- □ в) *while true or false do*
- □ г) *while (6 <= 6) and (5 >= 5) do*

9.9. Какие из высказываний не верны?
- □ а) Цикл *repeat* выполняется по меньшей мере один раз
- □ б) В цикле *repeat* не нужны скобки *begin/end*
- □ в) Цикл *repeat* должен иметь индексную переменную
- □ г) Цикл *repeat* может быть бесконечным

9.10. Какой из этих циклов бесконечен?
- □ а) *until true and false;*
- □ б) *until true or false;*
- □ в) *until (-2 < 6) and (3 > 2);*
- □ г) *until (-4 < -5) or (10 >= 10);*

9.11. Дано: *type Days = (Mon,Tue,Wed,Thu,Fri,Sat,Sun);*
- □ а) *Succ(Tue) = Wed*
- □ б) *Pred(Tue) = Mon*
- □ в) *Succ(Wed) = Pred(Fri)*
- □ г) Все выражения правильны

9.12. *SetVar := ['AA'..'ZZ'];*
- □ а) содержит 26 элементов
- □ б) содержит более 26 элементов
- □ в) содержит 0 элементов
- □ г) выражение не допустимо в Паскале

9.13. [1, 2, 3] * [4, 5, 6] =
- □ а) [1, 2, 3, 4, 5, 6]
- □ б) []
- □ в) [4, 5, 6]
- □ г) [1, 2, 3]

9.14. Copy('ABCDEFGHIJ', 4, 3) =
- □ а) 'CDEF'
- □ б) 'ABCD'
- □ в) 'DEF'
- □ г) Ответ другой

9.15. *Pos('DE', 'ABCDEFGH') =*
- □ а) 3
- □ б) 4
- □ в) 5

9.16. *Val('123', IntegerVar, ErrorCode);*
- □ а) *IntegerVar = 123, ErrorCode = 0*

□ г) 6

9.17. В чем состоит основное отличие между *write* и *writeln*?

□ а) *Write* может иметь только один аргумент

□ б) *Writeln* выводит еще и возврат каретки/перевод строки

□ в) *Writeln* ожидает нажатия клавиши '*Return*'

9.19. Какое значение следующего выражения?

```
var r: real;
m: integer;
r:=0.7;
m:=4;
round(r)+m+m*2
```

Ответ: _____.

9.21. Сколько раз будет выполнен следующий оператор цикла?

```
i:=1;
while i<= 10 do
i:=i+1;
i:=i+2;
```

Этот цикл будет выполняться _____ раз.

□ б) *IntegerVar = 123, ErrorCode <> 0*

□ в) *IntegerVar = 0, ErrorCode = 123*

□ г) Выражение не допустимо в Паскале

9.18. Какие процедуры не используются с файлами с произвольным доступом?

□ а) *Seek*

□ б) *BlockWrite*

□ в) *Write*

□ г) *Read*

9.20. Какое значение будет иметь переменная x после выполнения следующих операторов:

```
x:=0;
number1:=14;
number2:=5;
if number1>10 then
if number2<10
then x:=25
else x:=50;
```

Ответ: _____.

9.22. Сколько раз будет выполнен следующий оператор цикла?

```
m:=1;
repeat
m:=m+1;
m:=m+m
until m=10;
```

Этот цикл будет выполняться _____ раза.

9.23. Какой оператор будет выполнен:

```
sel1:=8;
case sel1 of
0,1,2: оператор1;
4,5,6: оператор2;
7,8,9: оператор3;
end;
```

□ а) оператор1

□ б) оператор2

□ в) оператор3

□ г) Никакой

9.25. Определить содержимое файла *f* после выполнения следующих операторов:

```
program prfile;var f: file of char;
s1,s2,s3,s4: char;beginassign(f,'f');
rewrite(f);s1:='1'; s2:='2'; s3:='3';
s4:='4';if eof(f) then write(f,s1) else
write(f,s2);if eof(f) then write(f,s3) else
write(f,s4)end.
```

□ а) 14

□ б) 13

□ в) 23

□ г) 24

9.24. Какое значение будет иметь переменная *x* после выполнения следующих операторов:

```
x:=0;
number1:=15;
number2:=11;
if number1>10 then
if number2<10
then x:=25
else x:=50;
```

Ответ: _____.

9.26. Пусть файл *f* содержит два числа: 3 и 7.

Определить, какое значение будет иметь переменная *y* после выполнения следующих операторов:

```
program prfile;varf: file of byte;x,y:
byte;beginassign(f,'f');
reset(f);y:=1;repeatread(f,x);y:=y*x
until eof(f)end.
```

Ответ: _____.

10. ТЕСТЫ ПО ТЕМЕ «РЕКУРСИЯ»

Ну? – повторял Ипполит Матвеевич на разные лады.
– Ну? Ну?
– Ну и ну, – отвечал Остап раздраженно, – один шанс против одиннадцати. И этот шанс...
Он хорошенько порылся в стуле и закончил:
– И **этот шанс пока не наш**.
Илья Ильф, Евгений Петров Двенадцать стульев.

10.1. Подпрограмма называется рекурсивной, если она вызывает саму _____. Такая рекурсия называется прямой.

10.2. Место хранения точек возврата называется _____ возврата.

10.3. Количество рекурсивных вызовов называется _____ рекурсии.

10.4. Рекурсивные алгоритмы особенно подходят для задач, где обрабатываемые данные определяются в терминах _____.

10.5. Мощность _____ определения заключается в том, что оно позволяет с помощью конечного высказывания определить бесконечное множество объектов.

10.6. При создании рекурсивных процедур рекомендуется компилировать программу с директивой {$___ +}. Эта директива включает проверку переполнения стека (область памяти, в которой хранится состояние вызывающей подпрограммы).

10.7. При каждой рекурсивной активации процедуры порождается новое множество _____ переменных.

10.8. В _____ возврата запоминаются не только адреса точек возврата, но и значения всех локальных переменных, образно выражаясь, запоминается слепок процедуры и функции.

10.9. Дана программа:

```
function fact(n:integer):longint;
begin
if n=1 then fact:=1 else
fact:=fact(n-1)*n;
end;
begin
writeln(fact(5));
end.
```

Что появится на экране компьютера после выполнения этой программы?
Ответ: _____

10.10. Дана программа:

```
Procedure Rec(n: integer);
Begin
If n> then Rec(n div 2);
Write(n mod 2);
End;
begin
write(Rec(6));
end.
```

Что появится на экране компьютера после выполнения этой программы?
Ответ: _____

10.11. Дана программа:

```
Procedure Rec(n: integer);
Begin
If n> then Rec(n div 2);
Write(n mod 2);
End;
begin
write(Rec(13));
end.
```

Что появится на экране компьютера после выполнения этой программы?

10.12. Дана программа:

```
Procedure Solve;
Var n:integer;
Begin Readln(n); if n<>0 then Solve;
Write(n) end;
begin Solve end.
```

Что появится на экране компьютера после ввода с клавиатуры последовательности чисел 1,2,3,0?

10.13. Дана программа:

```
function q(m:integer):integer;
begin
if m=1 then q:=1 else q:=q(m-
1)+m+m-1;
end;
begin
write(q(7));
end.
```

Что появится на экране компьютера после выполнения этой программы?
Ответ: _____

10.14. Являются ли данные стихи рекурсивными?

У попа была собака,
Он ее любил.
Она съела кусок мяса,
Он ее убил.
И в ямку закопал,
И надпись написал:
У попа была собака,
Он ее любил.
Она съела кусок мяса,
Он ее убил...

10.15. Если тело рекурсивной процедуры *P* имет вид:

begin
P;
операторы;
end;

То такая процедура реализует

□ а) рекурсивный спуск
□ б) рекурсивный подъем

10.17. Появление рекурсивных процедур стало возможным при динамическом распределении _____.

10.19. Дана программа:

```
Function K(N:Longint):Byte;
Begin
If N<10 Then
K:=1
Else
K:=K(N div 10)+1
End;
begin writeln(K(543210) end.
```

Что появится на экране компьютера после выполнения этой программы?
Ответ: _____.

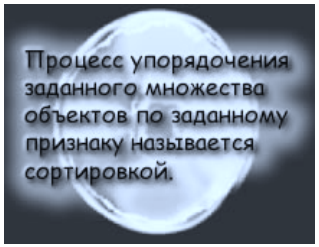
□ а) Да
□ б) Нет

10.16. Использование рекурсии увеличивает время исполнения программы и зачастую требует значительного объёма памяти для хранения копий подпрограммы на рекурсивном спуске. Поэтому на практике разумно заменять рекурсивные алгоритмы на _____.

10.18. Чтобы 'стек' экономно расходовал память, не следует вводить в теле рекурсивной процедуры много локальных _____, всегда не желательно определять массивы.

10.20. Обращение к рекурсивной подпрограмме ничем не отличается от вызова любой другой подпрограммы. При этом при каждом новом рекурсивном обращении в памяти создаётся новая копия подпрограммы со всеми локальными переменными. Такие копии будут порождаться до выхода на граничное условие. Очевидно, в случае отсутствия граничного условия, неограниченный рост числа таких копий приведёт к аварийному завершению программы за счёт переполнения _____.

11. ТЕСТЫ ПО ТЕМЕ «СОРТИРОВКА И ПОИСК»



Остап восторженно смотрел на старика.
– **Дивная канцелярия**, – сказал он, – **полная механизация**. Вы прямо герой! Польщенный архивариус стал вводить гостя в детали любимого дела.
Он раскрыл толстые книги учета и распределения.
– Все здесь, – сказал он, – весь Старгород! Вся мебель! У кого когда взято, кому когда выдано. А вот это – **алфавитная книга – зеркало жизни**!
Илья Ильф, Евгений Петров Двенадцать стульев.

11.1. Процесс упорядочения заданного множества объектов по заданному правилу называется _____.

11.3. Сложность сортировки вставками является

□ а) линейной
□ б) квадратичной
□ в) логарифмической

11.5. Элементы одномерного массива с именем a, содержащего n элементов, отсортированы по возрастанию, если каждый следующий элемент _____ предыдущего.

11.7. Массив до сортировки
30 17 73 47 22 11 65 54
После сортировки по возрастанию элементы станут равными _____

11.2. Укажите параметры, которые влияют на эффективность сортировки

□ а) простота программирования
□ б) красота названия
□ в) требуемое время
□ г) требуемая память

11.4. Сложность бинарного поиска является

□ а) линейной
□ б) квадратичной
□ в) логарифмической

11.6. Элементы одномерного массива с именем a, содержащего n элементов, отсортированы по невозрастанию, если каждый следующий элемент _____ предыдущего.

11.8. Количество проходов необходимых для сортировки методом 'пузырька' следующего (почти упорядоченного!) массива
5, 7, 12, 28, 36, 85, 2

11.9. Основной вопрос задачи поиска: где в заданной совокупности данных находится элемент, обладающий заданным _____?

11.11. Известно, что в целочисленном массиве A хранятся следующие значения

-1 1 2 3 5

Этот массив

□ а) не отсортирован
□ б) отсортирован по возрастанию
□ в) отсортирован по неубыванию
□ г) отсортирован по убыванию
□ д) отсортирован по невозрастанию

11.13.

```
procedure sort(n:integer; Var A: tarr);
var i,j,w: integer;
begin
for i:=2 to n do begin
w:=a[i]; i:=i-1;
while (j > 0) and (w < a[j]) do begin
a[j+1]:=a[j]; j:=j-1; end;
a[j+1]:=w end; end;
```

Эта процедура реализует алгоритм

□ а) сортировки простым обменом
□ б) сортировки простым выбором
□ в) сортировки простыми вставками
□ г) сортировка методом пузырька
□ д) сортировка методом подсчета

11.10. Метод сортировки обменом - это метод _____.

11.12. Известно, что в целочисленном массиве A хранятся следующие значения

1 3 -1 2 5

После первого прохода методом пузырька последовательность значений будет

□ а) 1 -1 2 3 5
□ б) 1 3 -1 2 5
□ в) 1 2 -1 3 5
□ г) -1 1 3 2 5
□ д) 1 -1 2 3 5

11.14.

```
procedure sort(n:integer; Var A: tarr);
var i,j,w: integer;
begin
for j:=1 to n-1 do
for i:=1 to n-j do
if a[i]>a[j+1] then begin w:=a[i];
a[i]:a[i+1]; a[i+1]:=w end;
end;
```

Эта процедура реализует алгоритм

□ а) сортировки простым обменом
□ б) сортировки простым выбором
□ в) сортировки простыми вставками
□ г) сортировка методом пузырька
□ д) сортировка методом подсчета

11.15. Известно, что в целочисленном массиве A хранятся следующие значения

1 3 -1 2 5

После второго прохода методом простого выбора последовательность значений будет

□ а) 1 -1 2 3 5
□ б) 1 3 -1 2 5
□ в) 1 2 -1 3 5
□ г) -1 1 3 2 5
□ д) 1 -1 2 3 5

11.17. Бинарный поиск это

□ а) двоичный поиск
□ б) дихотомический поиск
□ в) метод деления пополам
□ г) логарифмический поиск

□ а) сортировки простым обменом
□ б) сортировки простым выбором
□ в) сортировки простыми вставками
□ г) сортировка методом пузырька
□ д) сортировка методом подсчета

11.16.

```
procedure sort(n:integer; Var A: tarr);
var i,j,k,w: integer;
begin
for i:=1 downto 2 do begin
k:=i; w:=a[i];
for j:=1 to i-1 do if a[j]>w then begin
k:=j; w:=a[j] end;
if k<>i then begin a[k]:=a[i];
a[i]:=w; end; end; end;
```

Эта процедура реализует алгоритм

□ а) сортировки простым обменом
□ б) сортировки простым выбором
□ в) сортировки простыми вставками
□ г) сортировка методом пузырька
□ д) сортировка методом подсчета

11.18. Элементы одномерного массива с именем a, содержащего n элементов, отсортированы по невозрастанию, если между элементами выполняются неравенства

a[1] ? a[2] ? a[3] : ? a[n]

Вместо знака вопроса должен стоять знак

□ а) меньше
□ б) меньше или равно

11.19. Методы сортировки классифицируются на внутренние (когда данные размещаются в оперативной памяти) и _____ (когда данные размещаются на внешней памяти).

□ в) больше
□ г) больше или равно

11.20. Самые простые алгоритмы сортировки, основанные на сравнении элементов, имеют сложность порядка
□ а) $O(n^2)$
□ б) $O(n\log_2 n)$

12. ТЕСТЫ ПО ТЕМЕ «СРЕДСТВА ГРАФИКИ»

– Ну, Киса, – заметил Остап, – придется с утра сесть за работу.
Надеюсь, что вы сможете разводить краски.
А потом вот что: я **художник, окончил ВХУТЕМАС**, а вы мой помощник. Если вы думаете, что это не так, то скорее бегите назад, на берег.
Илья Ильф, Евгений Петров Двенадцать стульев.

12.1. Графические драйверы разработаны фирмой *Borland* практически для всех типов адаптеров. Обычно они располагаются на диске в отдельном подкаталоге в виде файлов с расширением
□ а) *gbi*
□ б) *bgi*
□ в) *ibg*
□ г) *gif*
□ д) *drv*

12.2. Процедура *InitGraph* инициирует графический режим работы адаптера. Заголовок процедуры:
Procedure InitGraph(var Driver, Mode: Integer; Path: String);
Здесь *Driver* - переменная типа *Integer*, определяет _____ графического драйвера;
Mode - переменная того же типа, задающая режим работы графического адаптера;
Path - выражение типа *String*, содержащее имя файла драйвера и, возможно, маршрут его поиска.

12.3. Если тип адаптера ПК неизвестен или если программа рассчитана на работу с любым адаптером, используется обращение к процедуре с требованием автоматического определения типа драйвера:
Driver :=_____;
InitGraph(Driver, Mode, 'C:\TP\BGI');

12.4. Функция, которая возвращает значение типа Integer, в котором закодирован результат последнего обращения к графическим процедурам(если ошибка не обнаружена, значением функции будет ноль, в противном случае - отрицательное число), имеет имя:
□ а) *ResultGraph*
□ б) *ResGraph*
□ в) *GraphResult*
□ г) *GraphRes*

12.5. Функция, которая завершает работу адаптера в графическом режиме и восстанавливает текстовый режим работы экрана, имеет имя:
□ а) *GrClose*
□ б) *CloseGr*
□ в) *GraphClose*
□ г) *CloseGraph*

12.6. В графическом режиме экран дисплея рассматривается как совокупность очень близко расположенных точек - _____, светимостью которых можно управлять с помощью программы.

12.7. Многие графические процедуры и функции используют указатель текущей позиции на экране, который в отличие от текстового курсора невидим. Положение этого указателя, как и вообще любая координата на графическом экране, задается относительно левого _____ угла, который, в свою очередь, имеет координаты 0,0.

12.8. Настройка графических процедур на работу с конкретным адаптером достигается за счет подключения нужного графического _____.

12.9. Функция, которая

12.10. В графическом режиме, из

возвращает значение типа *Word*, содержащие максимальную координату экрана, в текущем графическом режиме работы, по горизонтали имеет имя

□ а) *getx*
□ б) *getmaxx*
□ в) *gety*
□ г) *getmaxy*

12.11. В графическом режиме, оператор

Line(100,100,100,120);
нарисует отрезок

□ а) горизонтальный, длиной 20 пикселей
□ б) вертикальный, длиной 20 пикселей

12.13. В графическом режиме ось абсцисс направлена

□ а) снизу-вверх
□ б) справа-налево
□ в) слева-направо
□ г) сверху-вниз

12.15. Процедура, которая вычеркивает линию с указанными координатами начала и конца, имеет имя

□ а) *putpixel*
□ б) *lineto*
□ в) *linerel*
□ г) *line*
□ д) *rectangle*

12.17. Функция ImageSize возвращает размер памяти в байтах, необходимый для размещения прямоугольного фрагмента изображения. Заголовок:
Function ImageSize(X1,Y1, X2,Y2: Integer): Word;
Здесь *X1... Y2* - координаты левого верхнего (*X1, Y1*) и правого нижнего (*X2, Y2*) углов фрагмента
_____.

12.19. Функция, которая заполняет произвольную замкнутую фигуру, используя текущий стиль заполнения (узор и цвет, имеет имя

□ а) *FloodFill*
□ б) *Bar*
□ в) *Bar3D*
□ г) *Fill Poly*
□ д) *FillEllipse*
□ е) *Sector*

12.21. В графическом режиме минимальным объектом, отображаемым на экране, является

□ а) символ

перечисленных ниже операторов, размер экрана будут определять операторы

```
xm:=GetMaxX;
ym:=GetMaxY;
xc:=xm div 2;
yc:=ym div 2;
```

□ а) первый и четвертый
□ б) первый и третий
□ в) первый и второй
□ г) третий и четвертый
□ д) второй и третий

12.12. В графическом режиме, оператор который задает черный цвет линий имеет вид

□ а) *SetColor(0);*
□ б) *SetColor(black);*
□ в) *SetColor(dark);*
□ г) *SetColor(empty);*

12.14. Восстановите правильную последовательность ниже перечисленных операторов.

```
Error := GraphResult;
InitGraph(Driver, Mode,' ');
Driver := Detect;
if Error <> grOk then
WriteLn(GraphErrorMsg(Error));
```

Ответ: _____.

12.16. Приведенный ниже код:

```
Repeat
x:=Random(xm);y:=Random(ym);
n:=random(16);
IF(x>140)and(x<500)and(y>100)
and(y<350) then PutPixel(x,y,n)
Until KeyPressed;
```

рисует звездное небо

□ а) в круге
□ б) вне круга
□ в) вне прямоугольника
□ г) в прямоугольнике
□ д) вне эллипса
□ е) в эллипсе

12.18. Установите соответствие между файлами и их содержимым

1. Утроенный шрифт
2. Уменьшенный шрифт
3. Прямой шрифт
4. Готический шрифт

1. sans.chr
2. litt.chr
3. goth.chr
4. trip.chr

Ответ: _____.

12.20. Графическая функция, которая выводит строку, начиная с заданного места, имеет имя

□ а) *OutText*
□ б) *OutTextXY*

12.22. Библиотека _____ содержит в общей сложности более 50 процедур и функций, предоставляющих программисту самые разнообразные возможности

□ б) пиксел

12.23. Количество точек (пикселей), умещающихся на экране по вертикали и горизонтали, называют разрешающей _____.

12.25. Процедура *Circle* вычерчивает окружность. Заголовок: *Procedure Circle(X,Y: Integer; R: Word);*
Здесь *X, Y*- координаты _____; *R* - радиус в пикселях.

12.27. При работе с графикой Турбо Паскаля используются два вида шрифтов, различающихся своим внутренним форматом - растровые (он дин) и _____ (их несколько).

12.29. При написании программы, которая работает с графикой, обязательно должны присутствовать операторы и вызываться подпрограммы
□ а) *Circle*
□ б) *Uses Graph*
□ в) *InitGraph*
□ г) *Ellipse*
□ д) *SetColor*

управления графическим экраном.

12.24. Начало координат (0,0) в графическом режиме находится в
□ а) правом нижнем
□ б) левом верхнем
□ в) левом нижнем
□ г) правом верхнем
угле.

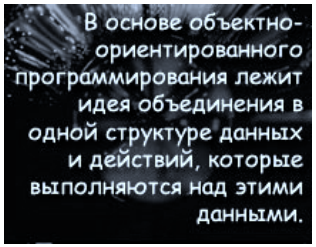
12.26. Функция, которая вычерчивает произвольную ломаную линию, заданную координатами точек излома, имеет имя:
□ а) *Box*
□ б) *Rectangle*
□ в) *Circle*
□ г) *DrawPoly*

12.28. Разрешающая способность экрана в режиме *VGAHi*
□ а) 480x640
□ б) 800x600
□ в) 640x480
□ г) 600x800
□ д) 350x640

12.30. Графические элементы, такие как точка, отрезок прямой линии, дуга, окружность и т.д. называются графическими _____.

□ е) *CloseGraph*

13. ТЕСТЫ ПО ТЕМЕ “ВВЕДЕНИЕ В ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ”



– Новая партия старушек? – спросил Остап.
– Это сироты, – ответил Альхен, выжимая плечом инспектора из кухни и исподволь грозя сиротам кулаком.
– Дети Поволжья?
Альхен замялся.
– Тяжелое **наследье** царского режима?
Илья Ильф, Евгений Петров Двенадцать стульев.

13.1. Выберите термин, который относится к объектно-ориентированному программированию
□ а) калькуляция
□ б) модуляция
□ в) инкапсуляция
□ г) аккумуляция
□ д) артикуляция
□ е) грануляция

13.3. Комбинирование записей с процедурами и функциями, манипулирующими полями этих записей, формирует новый тип данных - объект. Это
□ а) инкапсуляция
□ б) наследование
□ в) полиморфизм

13.5. Тип объекта может быть определен как полный, самостоятельный тип в манере

13.2. Выберите термин, который относится к объектно-ориентированному программированию
□ а) скирдование
□ б) доследование
□ в) заведование
□ г) собеседование
□ д) наследование
□ е) исследование

13.4. В терминах Паскаля, объект во многом схож с
□ а) переменной
□ б) записью
□ в) функцией
□ г) процедурой
□ д) константой
□ е) множеством

13.6. Порождающий объект называется
□ а) производительным

описания записей Паскаля, но он может определяться и как наследник существующего типа объекта путем помещения порождающего (родительского) типа в скобки после зарезервированного слова object. Эти скобки

□ а) квадратные
□ б) фигурные
□ в) круглые
□ г) косые

13.7. Одной из важных особенностей, которые объектно-ориентированное программирование добавляет традиционным языкам типа Паскаль, является механизм, с помощью которого типы данных могут наследовать характеристики более простых, более общих типов. Этим механизмом является _____.

13.9. Пусть дан тип-объект

```
type
TEmployee = object
Name, Title: string[25];
Rate: Real;
procedure Init (AName, ATitle: String;
ARate: Real);
function GetName : String;
function GetTitle : String;
function GetRate : Real;
end;
```

Определите сколько в нем методов

□ б) генезисным
□ в) декларативным
□ г) родительским
□ д) безродным
□ е) прародительским

13.8. Одним из важнейших принципов объектно-ориентированного программирования является то, что программист во время разработки программы должен думать о коде и о _____ совместно.

13.10. Пусть имеется описание

```
type
TEmployee = object
Name: string[25];
Title: string[25];
Rate : Real;
end;

THourly = object(TEmployee)
Time : Integer;
end;
```

Здесь *THourly* является

□ а) родительским типом

□ а) 7
□ б) 6
□ в) 5
□ г) 4
□ д) 3
□ е) 1

13.11. Если метод полностью определяется вне объекта, то имени метода должно предшествовать имя типа объекта, которому принадлежит этот метод. В качестве разделителя между этими имена выступает символ

□ а) собака
□ б) двоеточие
□ в) шляпка
□ г) точка
□ д) запятая
□ е) тире

13.13. Процесс, с помощью которого вызовы статических методов однозначно разрешаются компилятором во время компиляции в один метод, называется

□ а) ранним связыванием;
□ б) поздним связыванием.

13.15. Конструктор обычно присваивает полям объекта

□ б) дочерним типом

13.12. Полный синтаксис описания объекта имеет вид:

```
type
NewObject = object(родитель)
поля_1;
методы_1;
private
поля_2;
методы_2;
end;
```

Здесь поля_1 и методы_1 являются

□ а) общедоступными полями и методами
□ б) приватными полями и методами

13.14. Вызывающий и вызываемый методы могут быть не связаны во время компиляции, поэтому включается механизм, позволяющий осуществить связывание несколько позднее, когда вызов действительно произойдет. Такой процесс называется

□ а) ранним связыванием;
□ б) поздним связыванием.

13.16. Для выделения памяти переменной объектного типа

начальные значения и выполняет другие действия по инициализации объекта. В заголовке метода-конструктора слово procedure заменяется словом

□ а) *virtual*
□ б) *private*
□ в) *constructor*
□ г) *destructor*

13.17. Для очистки и удаления динамически размещенного объекта Турбо Паскаль предоставляет специальный тип метода, называемый 'сборщиком мусора' или

□ а) протектором
□ б) рефлектором
□ в) экстрактором
□ г) деструктором
□ д) инспектором
□ е) редактором

13.19. Любой объект может иметь сколько угодно потомков, но только одного _____, что позволяет создавать иерархические деревья наследования объектов.

13.21. К полю объекта можно обратиться в точности так же, как к полю обычной записи, либо с помощью оператора _____, либо путем уточнения имени с помощью точки.

13.23. Большая часть конструирования объектно-ориентированных прикладных программ состоит в построении такой иерархии объектов, являющейся отражением генеалогического _____ объектов в приложениях.

13.25. Отношения частей к целому и взаимоотношения между частями становятся понятнее тогда, когда все содержится вместе в одной упаковке. Это называется _____.

13.27. В ООП методы делятся на статические и в_____.

13.29. В ООП статический метод действует по принципу

□ а) Ты знаешь, чего я хочу. Пойди и вызови это.
□ б) Ты не знаешь пока, чего я хочу. Когда придет время, задай вопрос о конкретном экземпляре.

используется оператор *New(P,Constructor)*, который эквивалентен следующей паре операторов

□ а) *New(P); P^.Constructor;*
□ б) *P^.Destructor; Dispose(P);*

13.18. Зарезервированное слово *object* используется для описания объекта. Описание объекта должно помещаться в разделе описания типов:

```
type
MyObject = object
( _____ объекта}
{Методы объекта}
end;
```

13.20. Полями объекта могут быть любые структуры данных, в том числе и другие _____.

13.22. Метод объекта является процедурой или _____, описанной внутри объекта и жестко ограниченной этим объектом.

13.24. Экземпляры объектных типов описываются в точности так же, как в Паскале описывается любая переменная, либо статическая, либо как _____, ссылающийся на размещенную в динамической памяти переменную.

13.26. Объектно-ориентированное программирование в большой степени является процессом построения генеалогического дерева для структур _____.

13.28. По_____ - свойство различных объектов выполнять одно и то же действие (метод) по-своему.

13.30. В ООП виртуальный метод действует по принципу

□ а) Ты знаешь, чего я хочу. Пойди и вызови это.
□ б) Ты не знаешь пока, чего я хочу. Когда придет время, задай вопрос о конкретном экземпляре.

14. ТЕСТЫ ПО ТЕМЕ «ОСНОВЫ DELPHI»

– Вы знаете, сейчас в Европе и в лучших домах Филад**ельфи**и возобновили старинную моду – разливать чай через ситечко. Необычайно эффектно и очень элегантно.
Илья Ильф, Евгений Петров Двенадцать стульев.

14.1. Среда программирования Delphi состоит из ряда элементов, всегда присутствующих на экране:
- главного окна (*Project1*);
- окна инспектора объектов (*Object Inspector*);
- окна _____ (*Form1*);
- окна кода программы (*Unit1.pas*);
- окна проводника кода (*Code Explorer*).

14.3. Восстановите соответствие между окнами и их заголовками
А) Главное окно
Б) Окно инспекторов объектов
В) Окно формы
Г) Окно кода программы
Д) Окно проводника кода
1. *Form1*
2. *Unit1.pas*
3. *Delphi*
4. *Object Inspector*
5. *Code Explorer*
6. *Project1*
Ответ: _____.

14.2. Быстрое переключение между окнами формы и кода программы осуществляется нажатием клавиши
□ а) *F1*
□ б) *F6*
□ в) *F10*
□ г) *F12*

14.4. Количество форм, содержащееся в проекте равно
□ а) 1
□ б) 2
□ в) 3
□ г) 4
□ д) зависит от конкретной программы
□ е) не ограничено

14.5. Во время проектирования форма Delphi покрыта сеткой из точек. В узлах этой сетки будут находиться _____, помещаемые на форму.

14.7. Инспектор объектов позволяет устанавливать свойства объектов и назначать _____ (т.е. обработчики событий) во время создания программы.

14.9. В Delphi размер формы и ее положение на экране, а также размер и положение элементов управления на поверхности формы задаются в
□ а) сантиметрах
□ б) миллиметрах
□ в) пунктах
□ г) пикселах

14.11. В Delphi, текст, находящийся в форме, называют
□ а) именем
□ б) примечанием
□ в) меткой
□ г) обозначением
□ д) надписью

14.13. В Delphi, при наборе текста программы можно получить

14.6. Окно кода программы представляет собой текстовый _____, предназначенный для размещения текстов программных единиц проекта (программы-проекта, модулей, текстовых файлов) с целью их создания, подключения к проекту, редактирования.

14.8. Окно инспектора объектов содержит две вкладки: _____ и события.

14.10. В Delphi форма и ее компоненты рассматриваются как
□ а) переменные
□ б) константы
□ в) модули
□ г) объекты
□ д) массивы
□ е) указатели

14.12. В Delphi, при щелчке кнопкой мыши вызывается событие
□ а) *OnClick*
□ б) *OnDblClick*
□ в) *OnMouseDown*
□ г) *OnMouseMove*
□ д) *OnMouseUp*
□ е) *OnKeyDown*

14.14. Программирование, предусматривающее создание

справку о конструкции языка программирования, процедуре или функции. Для этого необходимо в окне редактора кода напечатать слово, о котором надо получить справку, затем установить курсор на любую его букву и нажать функциональную клавишу _____.

14.15. Двоичный файл, содержащий все необходимые для проекта (пиктограммы, курсора мыши, графические изображения или строки) называется файлом

□ а) ресурсов
□ б) опций
□ в) проекта
□ г) формы

14.17. Компонент, который показывает кнопку с надписью, называется

□ а) *Button*
□ б) *ComboBox*
□ в) *CheckBox*
□ г) *Edit*
□ д) *Label*
□ е) *ListBox*

14.19. В терминологии Delphi проект - это набор _____, используя которые компилятор создает файл исполняемой программы.

приложений с помощью наглядных средств называется

□ а) объектно-ориентированным программированием
□ б) визуальным программированием
□ в) case-программированием
□ г) структурным программированием

14.16. Файл, содержащий все ссылки на все формы проекта и относящиеся к ним модули, а также код инициализации приложения, называется файлом проекта. Он имеет расширение

□ а) *res*
□ б) *dof*
□ в) *dpr*
□ г) *dfm*
□ д) *pas*

14.18. При двойном щелчке на элементе формы появляется заготовка для процедуры, обрабатывающей событие, которое используется у компонента по у_____.

14.20. Двойной щелчок на изображении требуемого компонента в палитре компонент вызовет появление его на форме. При этом он будет размещен

14.21. В инспекторе свойств объектов для обозначения комбинированных свойств используется знак

□ а) -
□ б) +
□ в) !
□ г) *

14.23. Редактор кода Delphi автоматически выделяет _____ (зарезервированные) слова языка программирования Object Pascal (*begin, end, const, var* и другие) полужирным шрифтом, что делает текст программы более выразительным и облегчает восприятие структуры программы.

14.25. Регламентированный обмен информацией между человеком и компьютером, осуществляемый в реальном масштабе времени и направленный на совместное решение конкретной задачи: обмен информацией и координацией действий называется

□ а) пользовательский интерфейс
□ б) диалог
□ в) сообщение

14.27. Для того, чтобы создать процедуру обработки события, необходимо выполнить двойной

□ а) в верхнем левом угле
□ б) по центру
□ в) в нижнем правом угле

14.22. Пиктограммы позволяют быстро выполнять наиболее важные опции главного меню. Другими словами, щелчок на какой-либо из кнопок приводит к тому же результату, что и выбор соответствующей _____ главном меню.

14.24. Путем присвоения свойствам компонента определенных значений и связывания с обработчиками событий определенных процедур задаются функциональные возможности _____.

14.26. Порция информации, участвующая в диалоговом обмене называется

□ а) пользовательский интерфейс
□ б) диалог
□ в) сообщение

14.28. Палитра компонент позволяет выбрать компоненты, которые будут присутствовать в

щелчок в поле имени процедуры обработки _____.

14.29. _____ - любое воздействие от мыши или клавиатуры, либо действие, вызванное перерисовкой.

приложении. _____ - это строительные блоки, из которых собирается Delphi-приложение.

14.30. Создание приложения в Delphi подразумевает создание _____, размещение на ней необходимых компонентов, определение свойств (средства доступа к компонентам) и описание методов (функции и процедуры, выполняющие действия над компонентами) этих компонентов.

15. ТЕСТЫ ПО ТЕМЕ «ЯЗЫК OBJECT PASCAL»

– Теперь вы дозрели и приобрели полную возможность зарабатывать деньги честным трудом.
– Что же я должен делать? – слезливо спросил Воробьянинов.
– Французский **язык знаете, надеюсь**?
– Очень плохо. В пределах гимназического курса.
– Гм... Придется орудовать в этих пределах.
Илья Ильф, Евгений Петров Двенадцать стульев.

15.1. Основным инструментом программирования в Delphi является
□ а) *Free Pascal*
□ б) *TMT Pascal*
□ в) *Object Pascal*
□ г) *Turbo Pascal*
□ д) *Vitual Pascal*

15.2. Программа в Delphi состоит из файла _____ и одного или нескольких модулей.

15.3. Программа в Object Pascal может содержать следующие разделы:
заголовок программы

радел объявления подключенных модулей
раздел объявления меток
раздел объявления к
раздел объявления типов
раздел объявления переменных
раздел процедур и _____
тело программы (обязательная часть)

15.4. Зарезервированные слова в Object Pascal еще называются:
□ а) лечебными
□ б) служебными
□ в) целебными
□ г) судебными
□ д) врачебными
□ е) волшебными

15.5. В алфавит языка Object Pascal входят
□ а) цифры
□ б) буквы
□ в) шестнадцатеричные цифры
□ г) специальные символы
□ д) пробелы
□ е) зарезервированные слова

15.6. Отметьте, какие из перечисленных букв являются строчными
□ а) *P*
□ б) *a*
□ в) *s*
□ г) *c*
□ д) *a*
□ е) *l*

15.7. Прописные буквы еще называются
□ а) анклавными
□ б) булавными
□ в) тщеславными
□ г) заглавными
□ д) принципиальными
□ е) капительными

15.8. Количество строчных букв в латинском алфавите равно
□ а) 25
□ б) 26
□ в) 27
□ г) 28
□ д) 29

15.9. Специальные символы используются для конструирования (отметьте все варианты)
□ а) знаков операций
□ б) выражений
□ в) служебных слов
□ г) комментариев
□ д) синтаксических разделителей

15.10. Количество зарезервированных слов в Object Pascal порядка
□ а) 10
□ б) 100
□ в) 300
□ г) 1000

□ е) зарезервированных слов

15.11. Программист имеет право создавать

□ а) идентификаторы
□ б) служебные слова
□ в) процедуры
□ г) функции
□ д) директивы

15.13. В окне кода Delphi зарезервированные слова и директивы выделяются

□ а) подчеркиванием
□ б) другим цветом
□ в) жирным шрифтом
□ г) наклонным шрифтом
□ д) мерцающим шрифтом

15.15. В Object Pascal идентификаторы вводятся для обозначения (отметьте все варианты)

□ а) объектов
□ б) программ
□ в) служебных слов
□ г) директив
□ д) полей записей
□ е) типов

15.17. Идентификатор формируется из букв и цифр и

□ а) начинается всегда с цифры
□ б) начинается всегда с буквы
□ в) может начинаться с буквы или цифры

15.12. В программе на Object Pascal несколько пробелов, поставленных подряд (если это только не строковая константа) считаются

□ а) несколькими пробелами
□ б) одним пробелом

15.14. В Object Pascal идентификаторы вводятся для обозначения (отметьте все варианты)

□ а) переменных
□ б) констант
□ в) меток
□ г) процедур
□ д) классов
□ е) зарезервированных слов

15.16. Определение идентификатора

□ а) должно предшествовать его использованию
□ б) может находиться в любом месте

15.18. Длина идентификатора может быть произвольной, однако компилятор воспринимает только

□ а) первые 23 его символа
□ б) первые 47 его символов
□ в) первые 63 его символа

□ г) начинается всегда с символа подчеркивания

15.19. Идентификатор может содержать

□ а) пробелы
□ б) буквы
□ в) символ подчеркивания
□ г) специальные символы
□ д) цифры

15.21. Соответствующие заглавные и строчные буквы в идентификаторах и служебных словах

□ а) различаются
□ б) не различаются

15.23. Переменные в Object Pascal являются информационными объектами, предназначенными для хранения значений определенного

_____

15.25. В качестве констант могут использоваться

□ а) целые числа
□ б) вещественные числа
□ в) шестнадцатеричные числа
□ г) логические константы
□ д) символы
□ е) строки

15.27. Вещественные числа

□ г) первые 127 его символов
□ д) первые 255 его символов

15.20. Какие из перечисленных ниже комментариев не допустимы в языке Object Pascal

□ а) // однострочный комментарий
□ б) {однострочный комментарий }
□ в) (* Многострочный комментарий *)
□ г) {Вложенный{ комментарий}}
□ д) {Вложенный(*комментарий*)}

15.22. Текст комментариев

□ а) игнорируется компилятором
□ б) не изменяет смысл программы
□ в) не влияет на ее выполнение

15.24. Описание констант начинается с ключевого слова

□ а) *program*
□ б) *object*
□ в) *const*
□ г) *var*
□ д) *type*
□ е) *uses*

15.26. В Object Pascal числа могут быть

□ а) кардинальными
□ б) целыми
□ в) комплексными
□ г) вещественными
□ д) действительными

15.28. Шестнадцатеричное число

записываются в форме с десятичной точкой или в форме с использованием десятичного порядка, который изображается буквой

□ а) *d*
□ б) *D*
□ в) *E*
□ г) *e*
□ д) *P*
□ е) *p*

15.29. Логическая константа принимает

□ а) одно значение
□ б) два значения
□ в) три значения
□ г) количество значений, определяемое программистом

15.31. Использование в программе идентификаторов констант вместо записей конкретных значений

□ а) хорошим стилем программирования

□ б) делает программу более читабельной
□ в) способствует лучшему ее пониманию без какого-то то ни было снижения эффективности

15.33. Восстановите

состоит из шестнадцатеричных цифр, которым предшествует знак

□ а) $
□ б) H
□ в) доллара
□ г) #
□ д) №

15.30. Строковая константа в Object Pascal - это последовательность символов, записанная между

□ а) знаками амперсанда
□ б) знаками процента
□ в) знаками фигурных скобок
□ г) восклицательными знаками
□ д) апострофами
□ е) знаками доллара

15.32. Все операции можно разбить на следующие группы:
арифметические
логические
операции с битами информации
операции со ст_____
операции отношения
операции с множествами
адресная операция

15.34. Существенных свойством

соответствия между стандартными операциями в Object Pascal и их описанием

А) *mod*
Б) *and*
В) *shl*
Г) *shr*
Д) +

1. получение адреса
2. сложение
3. целочисленное деление
4. остаток от деления
5. умножение
6. отрицание
7. деление
8. сдвиг влево
9. сдвиг вправо
10. логическое и

Ответ:_____.

15.35. К структурированному типу данных относятся:

м_____
записи
множества
файлы

15.37. В языке Object Pascal по сравнению с Turbo Pascal добавлены следующие типы данных:

□ а) классы
□ б) указатели
□ в) варианты
□ г) дата-время

для типа данных в Object Pascal является:

□ а) название типа данных
□ б) множество значение, которое может принимать переменная этого типа
□ в) множество операций, допустимых над переменными данного типа
□ г) место в иерархии данных

15.36. К порядковому типу данных относятся:

целые
л_____
символьные
перечисляемые
тип-диапазон

15.38. Порядковый номер значений возвращает функция

□ а) *odd(x)*
□ б) *succ(x)*
□ в) *pred(x)*
□ г) *ord(x)*

□ д) объекты

15.39. Тип дата-время используется для хранения даты и _____.

15.41. Для символьного типа функция порядкового номера *ord* возвращает числа в диапазоне от _____ до _____.

15.43. Для целого числа *x* *ord(x)*= _____

15.45. Для порядкового типа справедливо равенство *ord(pred(x)) = _____(x) - 1*

15.47. В Object Pascal для хранения целого типа данных *Cardinal* резервируется

□ а) 1 байт
□ б) 2 байта
□ в) 4 байта
□ г) 8 байта

15.49. В Object Pascal к вещественным числам с плавающей точкой относятся

□ а) *Real*
□ б) *Single*
□ в) *Double*

15.40. Каждый из порядковых типов имеет конечное число _____.

15.42. Для перечисляемого типа функция порядкового номера *ord* возвращает числа в диапазоне от _____ до _____.

15.44. Все значения любого порядкового типа образуют упорядоченную последовательность, и значение переменной порядкового типа определяется его местом в этой _____.

15.46. В Object Pascal для хранения целого типа данных SmalltInt резервируется

□ а) 1 байт
□ б) 2 байта
□ в) 4 байта
□ г) 8 байта

15.48. В Object Pascal функция, которая преобразует строчную букву в прописную, имеет имя _____.

15.50. В Object Pascal для типа дата-время резервируется

□ а) 4
□ б) 6
□ в) 8
□ г) 10

байт

□ г) *Extended*
□ д) *Currency*
□ е) *Comp*

15.51. В Object Pascal функция, которая инициализирует генератор случайных чисел, имеет имя _____.

15.53. Тело программы представляет собой последовательность _____.

15.55. В Object Pascal к простым операторам относятся

□ а) оператор присваивания
□ б) вызов процедуры
□ в) оператор безусловного перехода
□ г) пустой оператор
□ д) составной оператор
□ е) условный оператор *if*
□ ж) условный оператор *case*
□ з) оператор цикла *repeat*
□ и) оператор цикла *while*
□ й) оператор цикла *for*
□ к) оператор над записями *with*

15.57. Массив - это представляет собой определенное (фиксированное) количество элементов одного _____, снабженных индексами.

15.52. В Object Pascal для того чтобы вычислить гиперболический тангенс можно воспользоваться функцией _____(x), но при этом необходимо указать в разделе *uses* модуль с именем *math*.

15.54. _____ - это единое неделимое предложение, выполняющее какое-либо алгоритмическое действие.

15.56. В Object Pascal к структурированным операторам относятся

□ а) оператор присваивания
□ б) вызов процедуры
□ в) оператор безусловного перехода
□ г) пустой оператор
□ д) составной оператор
□ е) условный оператор *if*
□ ж) условный оператор *case*
□ з) оператор цикла *repeat*
□ и) оператор цикла *while*
□ й) оператор цикла *for*
□ к) оператор над записями *with*

15.58. Размер массива в Object Pascal не должен превышать

□ а) 2 Kb
□ б) 2 Mb
□ в) 2 Gb
□ г) 2 Tb

15.59. Количество элементов в многомерном массиве равно произведению количества элементов во всех _____.

15.61. В Object Pascal операции отношения над элементами массивами
□ а) определены
□ б) не определены

15.63. В Object Pascal для освобождения памяти, отведенной под динамической памяти, достаточно имени массива присвоить значение
□ а) #0
□ б) Nil
□ в) ' '
□ г) Empty

15.65. В Object Pascal множество задается с помощью ключевых слов
□ а) *set are*
□ б) *set of*
□ в) *set by*
□ г) *of set*

15.67. В Object Pascal элементы множества
□ а) нельзя
□ б) можно
отобразить на экране компьютера

15.69. В Object Pascal строки

15.60. _____ массива хранятся в памяти последовательно друг за другом.

15.62. В Object Pascal можно определять динамические массивы, при этом необходимо
□ а) задать тип элементов
□ б) задать размерность (количество индексов)
□ в) задать количество элементов

15.64. В Object Pascal множество, в котором нет элементов, называется
□ а) незаполненным
□ б) пустым
□ в) безэлементным
□ г) потерянным
□ д) начальным
□ е) нулевым
□ ж) инфинитоземальным

15.66. В Object Pascal для задания множества используется перечисление значений в
□ а) круглых
□ б) квадратных
□ в) фигурных
скобках

15.68. В Object Pascal доступ к полям записи осуществляется путем указания имени переменной и имени _____ через точку.

15.70. В Object Pascal функция,

бывают четырех типов
1. *shortString* или *string[n]*, n< 256
2. длинная строка *String*
3. широкая строка *WideString*
4. строка, оканчивающаяся нулевым кодом _____

15.71. В Object Pascal есть возможность по ходу выполнения программы выделять и освобождать необходимую память для размещения в ней различных данных. Такая память называется
□ а) управленческой
□ б) технической
□ в) физической
□ г) динамической
□ д) обновленческой
□ е) экономической

15.73. В Object Pascal вся динамическая память рассматривается как сплошной массив байтов и называется
□ а) ящиком
□ б) кучей
□ в) кагатом
□ г) стогом
□ д) копной

15.75. В Object Pascal для объявления типизированного указателя перед базовым типом ставится знак указателя

которая конвертирует символы строки в вещественное число, имеет имя
□ а) *StrToInt*
□ б) *StrToTime*
□ в) *StrToFloat*
□ г) *StrToDate*

15.72. В Object Pascal работа с динамической памятью позволяет эффективно использовать оперативную память компьютера. Такая возможность связана с наличием типа данных
□ а) списки
□ б) очереди
□ в) стеки
□ г) отсылки
□ д) стрелки
□ е) персты
□ ж) указатели
□ з) сажени

15.74. В Object Pascal обычно указатель связывается с некоторым типом данных. Этот тип данных называется базовым, а указатель
□ а) монотипным
□ б) автотипным
□ в) типизированным
□ г) стереотипным
□ д) типическим

15.76. В Object Pascal чтобы получить указатель, не связанный ни с каким конкретным типом данных, необходимо воспользоваться стандартным типом

□ а) ^
□ б) @
□ в) *$*
□ г) *&*
□ д) *%*

15.77. В Object Pascal для того чтобы получить значение, с которым связан указатель, необходимо после имени указателя поставить знак
□ а) @
□ б) *&*
□ в) ^
□ г) *%*
□ д) *

15.79. В Object Pascal тип, который может динамически изменять свой тип, носит название
□ а) странный
□ б) свободный
□ в) вариативный
□ г) инвариантный
□ д) вариантный
□ е) дилемный
□ ж) неустойчивый
□ з) адаптивный
□ и) каркасный
□ й) мутирующий
□ к) нонконформистский

15.81. В Object Pascal при работе с динамической памятью следует следовать такому правилу: освобождать ровно столько памяти,

□ а) *list*
□ б) *stream*
□ в) *street*
□ г) *pointer*

15.78. В Object Pascal указатель называется неопределенным если его значение равно
□ а) *0*
□ б) *nil*
□ в) *empty*
□ г) *null*
□ д) *disannul*

15.80. В Object Pascal функция, которая выделяет динамическую область памяти для размещения переменной и возвращает в типизированный указатель адрес первого байта этой памяти, имеет имя
□ а) *Dispose*
□ б) *GetMem*
□ в) *AllocMem*
□ г) *SizeOf*
□ д) *New*
□ е) *FreeMem*
□ ж) *Addr*
□ з) *Ptr*

15.82. В Object Pascal _____ константы - это объявленные в разделе констант переменные, которым присвоены

сколько выделено, и именно с того _____, с которого она была выделена.

15.83. В Object Pascal функция, которая освобождает динамическую область памяти, выделенную под типизированный указатель, имеет имя
□ а) *Dispose*
□ б) *GetMem*
□ в) *AllocMem*
□ г) *SizeOf*
□ д) *New*
□ е) *FreeMem*
□ ж) *Addr*
□ з) *Ptr*

15.85. В Object Pascal существует две разновидности подпрограмм - процедуры и _____.

15.87. В Object Pascal отрицательным моментом использования переменных вариантного типа является то, что для каждой из них требуется дополнительная память, равная
□ а) 1 байту
□ б) 2 байта
□ в) 3 байта
□ г) 4 байта
(для хранения информации о

начальные значения.

15.84. В Object Pascal тип, который может динамически изменять свой тип, имеет имя
□ а) *freetype*
□ б) *typefree*
□ в) *freedom*
□ г) *variant*
□ д) *alltype*
□ е) *typeall*

15.86. В Object Pascal _____ - последовательность элементов одного типа, обычно расположенных на внешнем устройстве(жестком диске, дискете, диске CD-ROM).

15.88. В Object Pascal при описании подпрограмм обязательно наличие:
заголовка подпрограммы
раздела объявления меток
раздела объявления констант
раздела объявления типов
раздела объявления переменных
раздела процедур и функций
_____ подпрограммы

действительном типе переменной). Работа с вариантами выполняется намного медленнее.

15.89. В Object Pascal обращение к подпрограмме осуществляется через упоминание ее имени и называется

□ а) селекцией
□ б) посылкой
□ в) манифестом
□ г) вызовом
□ д) призывом
□ е) эмиссией
□ ж) переходом

15.91. В Object Pascal тип стандартного текстового файла имеет имя

□ а) *text*
□ б) *filetext*
□ в) *textfile*
□ г) *fileoftext*
□ д) *textoffile*

15.93. В Object Pascal доступ к файлам из программы возможен только после реализации процедуры открытия файла. В этой процедуре производится привязка ранее объявленной файловой переменной к имени, существующего или созданного файла. Эту операцию выполняет процедура

□ а) *Assign*
□ б) *AssigntoFile*
□ в) *AssignFile*

15.90. В Object Pascal подпрограмма может вызывать саму себя, и это называется

□ а) рекуперацией
□ б) рекультивацией
□ в) абберацией
□ г) иллюстрацией
□ д) рекурсией
□ е) декламацией

15.92. В Object Pascal стандартный тип textfile определяет файл, который содержит символы, объединенные в

□ а) столбцы
□ б) записи
□ в) строки
□ г) массивы

15.94. В Object Pascal для того, чтобы типизированная константа могла использоваться наравне с обычными переменными даже в левой части операции присваивания необходимо включить директиву компилятора

□ а) *{$J-}*
□ б) *{$I+}*
□ в) *{$S+}*
□ г) *{$J+}*
□ д) *{$R+}*

□ г) *FileAssign*

15.95. В Object Pascal установите назначения процедур и функций для работы с типизированными файлами

А) *FilePos(F)*
Б) *FileSize(F)*
В) *Read(F,список_ввода)*
Г) *Seek(F,Num)*
Д) *Truncate(F)*
Е) *Write(F,список_вывода)*

1. Записать информацию в файл
2. Удалить часть файла, начиная с текущей позиции
3. Переместить указатель на элемент с указанным номером Num
4. Прочитать информацию из файла
5. Вернуть текущий размер файла
6. Определить номер текущего элемента

Ответ:

_____.

15.97. В Object Pascal все указатели имеют одинаковый размер, равный

□ а) 2 байта
□ б) 4 байта
□ в) 6 байт
□ г) 8 байт
□ д) 10 байт

15.99. В Object Pascal при работе с динамическими массивами функция, которая возвращает самый маленький индекс массива, имеет имя

15.96. В Object Pascal установите назначения процедур и функций для работы с файлами без типа

А) *BlockRead(F,Buf,N[,R])*
Б) *BlockWrite(F,Buf,N[,R])*
В) *FilePos(F)*
Г) *FileSize(F)*
Д) *Seek(F,Num)*
Е) *Truncate(F)*

1. Прочитать элементы из файла
2. Записать элементы в файл
3. Получить номер текущего элемента
4. Вычислить текущий размер файла
5. Настроить указатель на требуемый элемент
6. Удалить часть файла, начиная с текущей позиции

Ответ:

_____.

15.98. В Object Pascal функция, которая конвертирует целое число в строку символов, имеет имя

□ а) *IntToStr*
□ б) *TimeToStr*
□ в) *FloatToStr*
□ г) *DateToStr*

15.100. При задании типа запись после зарезервированного слова следует перечислить все поля с указанием через двоеточие их типов и завершить задание типа слово

_____.

□ а) *length*
□ б) *higth*
□ в) *low*

СПИСОК ИСПОЛЬЗОВАННОЙ И РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию.– М.: Наука. Гл. ред. физ.-мат. лит., 1988.– 224 с.
2. Акимов О.Е. Дискретная математика: логика, группы, графы.– М.: Лаборатория Базовых Знаний, 2001.– 376 с.
3. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов.– М., Мир, 1979.– 536 с.
4. Бешенков С.А. Моделирование и формализация. Методическое пособие/С.А. Бешенков, Е.А. Ракитина.– М.: Лаборатория Базовых Знаний, 2002.– 336 с.
5. Бондарев В.М., Рублинецкий В.И., Качко Е.Г. Основы программирования.– Харьков: Фолио; Ростов н/Д: Феникс, 1998.– 368 с.
6. Брегель Ю.Э. Пословицы и поговорки народов Востока. – М., Издательство восточной литературы, 1961. – 736 с. (с.409, п.133)
7. Брукс Ф. Мифический человеко-месяц или как создаются программные системы.– Пер. с англ.– СПб.: Символ-Плюс, 1999.– 304 с.
8. Вирт Н. Систематическое программирование. Введение.– М., Мир, 1977.– 189 с.
9. Вирт Н. Алгоритмы и структуры данных: Пер. с англ.– СПб.: Невский Диалект, 2001.– 352 с.
10. Глушаков С.В., Клевцов А.Л., Терербилов С.А. Программирование на Delphi 5.0.– Харьков: Фолио, 2002.– 518 с.
11. Грогоно П. Программирование на языке Паскаль: Пер. с англ.– М.: Мир, 1982.– 384 с.
12. Гуденко Д.А., Петроченко Д.В. Сборник задач по программированию.– СПб.: Питер, 2003.– 475 с.

13. Дал У., Дейкстра Э., Хоор К. Структурное программирование: Пер. с англ.– М.: Мир, 1975.– 248 с.
14. Долинский М.С. Алгоритмизация и программирование на Turbo Pascal: от простых до олимпиадных задач : Учебное пособие.– СПб.: Питер, 2005.– 237 с.
15. Епанешников А., Епанешников В. Программирование в среде Turbo Pascal 7.0.–М.: Диалог-МИФИ, 1997.– 288 с.
16. Иванов Б.И. Дискретная математика. Алгоритмы и программы: Учеб. Пособие.– М.: Лаборатория Базовых Знаний, 2002.– 288 с.
17. Йодан Э. Структурное проектирование и конструирование программ.– М., Мир, 1979.– 416 с.
18. Караванова Т.П. Основи алгоритмізації та програмування: 750 задач з рек. та прикл.: Посіб.– К.: Форум, 2002.– 287 с.
19. Кристтофидес Н. Теория графов. Алгоритмический подход.– М., Мир, 1978.– 432 с.
20. Милов А.В. Основы программирования в задачах и примерах: Учебный курс.– Харьков: Фолио, 2002.– 397 с.
21. Немнюгин С.А. Turbo Pascal: практикум.– СПб.: Питер, 2003.– 256 с. (Серия «Учебники для вузов»)
22. Немнюгтн С.А. Turbo Pascal.– СПб.: Питер, 2002.– 496 с.
23. Окулов С.М. Основы программирования.– М.: ЮНИМЕДИАСТАЙЛ, 2002.– 424 с.
24. Окулов С.М. Программирование в алгоритмах.– М.: БИНОМ. Лаборатория знаний, 2002.– 341 с.
25. Павлидис Т. Алгоритмы машинной графики и обработки изображений: Пер. с англ.– М.: Радио и связь, 1986.– 400 с.
26. Перколаб Л.В., Муравка А.С. Основы программирования. Паскаль: примеры, задачи, решения.– Харьков: Гимназия, 200.– 168 с.
27. Пильщиков В.Н. Сборник задач по языку Паскаль: Учеб. Пособие для вузов.– М.: Наука. Гл. ред. физ.-мат. лит., 1989.– 160 с.
28. Роджерс Д. Алгоритмические основы машинной графики: Пер. с англ.– М.: Мир, 1989.– 512 с.
29. Тассел Д. Ван Стиль, разработка, эффективность, отладка и испытание программ: Пер. с англ.– М.: Мир, 1985.– 332 с.
30. Ускова О.Ф. Программирование на языке Паскаль: задачник.– СПб.: Питер, 2002.– 336 с.
31. Фаронов В.В. Турбо Паскаль 7.0 Начальній курс. Учебное пособие.– М.: Нолидж, 1997.– 616 с.

32. Юркин А. Задачник по программированию.– СПб.: Питер, 2002.– 192 с.

Навчальне видання

САВЧЕНКО Микола Володимирович
КУХАРЕНКО Володимир Миколайович

Збірник тестів до курсу
“Основи інформаційних технологій та програмування”

Навчально-методичний посібник
для студентів фізико-технічних спеціальностей

Російською мовою

Роботу до видання рекомендував О.П. Сук

За авторською редакцією
Зав. редакційно-видавничим
відділом М.П. Єфремова

План 2006 р., п. 10/177-05

Підп. до друку 26.12.2005 р. Формат 60x84 1/16. Папір офсетний. Riso-друк.
Гарнітура Таймс. Ум. друк. арк. 3,5. Обл.-вид. арк.4,4.
Наклад 50 прим. Зам. № 407. Ціна договірна.

Видавничий центр НТУ “ХПІ”.
Свідоцтво про державну реєстрацію ДК №116 від 10.07.2000 р.
61002, Харків, вул. Фрунзе, 21.

Друкарня НТУ “ХПІ”. 61002, Харків, вул. Фрунзе, 21.