


Федеральное государственное образовательное бюджетное учреждение
высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

Красноярский филиал Финуниверситета

(наименование структурного подразделения)

УТВЕРЖДАЮ
Заместитель директора по
учебно-методической работе
Красноярского филиала
Финуниверситета
 О.С. Вергейчик
« 04 » сентября 2025 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебному предмету / дисциплине

ОП.04 «Основы алгоритмизации и программирования»

(наименование учебного предмета/ дисциплины)

09.02.07 Информационные системы и программирование

(код, наименование специальности)

Красноярск – 2025 г.

Фонд оценочных средств по учебному предмету / дисциплине разработан на основании федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 «Информационные системы и программирование»

Составители:

Шадрин Валерий Георгиевич, канд. техн. наук, преподаватель ВКК

(фамилия, имя, отчество, наименование должности, квалификационной категории)

(фамилия, имя, отчество, наименование должности, квалификационной категории)

Фонд оценочных средств по учебному предмету / дисциплине рассмотрен и рекомендован к утверждению на заседании предметной (цикловой) комиссии общепрофессиональных дисциплин

(наименование)

Протокол от «04» сентября 2025 г. № 1

Председатель предметной (цикловой)
комиссии


(подпись)

О.А. Полтавец
(инициалы, фамилия)

1. Паспорт фонда оценочных средств
по учебному предмету / дисциплине
«Основы алгоритмизации и программирования»
(наименование)

09.02.07 Информационные системы и программирование

(код, наименование специальности)

Результаты обучения (знания, умения)	Общие и профессиональные компетенции	Наименование темы	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
<p>Усвоенные знания: Эволюцию языков программирования, их классификацию, понятие системы программирования.</p> <p>Освоенные умения: разрабатывать алгоритмы для конкретных задач.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 1.1. Языки программирования	1.Выполнение и защита практических- лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы и задания на написания программ для промежуточной аттестации в форме экзамена
<p>Усвоенные знания: Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>Освоенные умения: разрабатывать алгоритмы для конкретных задач.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 1.2.Типы данных.	1.Выполнение и защита практических- лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы и задания на написания программ для промежуточной аттестации в форме экзамена
<p>Усвоенные знания: Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</p> <p>Освоенные умения: Использовать программы для графического</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 2.1. Операторы языка программирования	1.Выполнение и защита практических- лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена

отображения алгоритмов.				
<p><u>Усвоенные знания:</u> Подпрограммы, составление библиотек подпрограмм</p> <p><u>Освоенные умения:</u> владение навыками и опытом разработки программ в выбранной среде программирования, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ; владение универсальным языком программирования высокого уровня (по выбору), представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 3.1. Процедуры и функции	1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена
<p><u>Усвоенные знания:</u> общие принципы построения алгоритмов, основные алгоритмические конструкции.</p> <p><u>Освоенные умения:</u> Работать в среде программирования.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 3.2. Структуризация в программировании на языке Pascal ABC	1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена
<p><u>Усвоенные знания:</u> Подпрограммы, составление библиотек подпрограмм</p> <p><u>Освоенные умения:</u> Работать в среде программирования.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 3.3. Модульное программирование	1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена

<p><u>Усвоенные знания:</u> Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p><u>Освоенные умения:</u> Работать в среде программирования.</p>	<p>ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5</p>	<p>Тема 4.1 Указатели .</p>	<p>1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование</p>	<p>Примерные вопросы для промежуточной аттестации в форме экзамена</p>
<p><u>Усвоенные знания:</u> объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.</p> <p><u>Освоенные умения:</u> Работать в среде программирования. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Оформлять код программы в соответствии со стандартом кодирования.</p>	<p>ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5</p>	<p>Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)</p>	<p>1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование</p>	<p>Примерные вопросы для промежуточной аттестации в форме экзамена</p>
<p><u>Усвоенные знания:</u> Подпрограммы, составление библиотек подпрограмм</p> <p><u>Освоенные умения:</u> владение опытом построения и использования компьютерно-математических моделей, проведения</p>	<p>ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5</p>	<p>Тема 5.2 Интегрированная среда разработки.</p>	<p>1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование</p>	<p>Примерные вопросы для промежуточной аттестации в форме экзамена</p>

экспериментов и статистической обработки данных с помощью компьютера, интерпретации результатов, получаемых в ходе моделирования реальных процессов; умение оценивать числовые параметры моделируемых объектов и процессов, пользоваться базами данных и справочными системами.				
<p>Усвоенные знания: Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>Освоенные умения: Работать в среде программирования. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования. Оформлять код программы в соответствии со стандартом кодирования.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09	Тема 5.3. Визуальное событийно-управляемое программирование в Pascal ABC	1.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена
<p>Усвоенные знания: Подпрограммы, составление библиотек подпрограмм</p> <p>Освоенные умения: использовать соответствующие графические редакторы для создания изображений.</p>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5	Тема 5.4 Разработка оконного приложения	11.Выполнение и защита практических-лабораторных работ, включая написание программ и контрольный опрос 2.Тестирование	Примерные вопросы для промежуточной аттестации в форме экзамена
Усвоенные знания: Основные элементы языка, структуру программы, операторы и операции, управляющие	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09	Тема 5.5 Этапы разработки приложений	1.Выполнение и защита практических-лабораторных работ, включая	Примерные вопросы для промежуточной аттестации в форме экзамена

<p>структуры, структуры данных, файлы, классы памяти.</p> <p>Освоенные умения:</p> <p>Выполнять проверку, отладку кода программы. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p>	<p>ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5</p>		<p>написание программ и контрольный опрос</p> <p>2.Тестирование</p>	
<p>Усвоенные знания:</p> <p>Основные классы памяти.</p> <p>Освоенные умения:</p> <p>Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p> <p>Выполнять проверку, отладку кода программы.</p>	<p>ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ПК 1.1- ПК 1.5, ПК 2.4, ПК 2.5</p>	<p>Тема 5.6</p> <p>Иерархия классов (наследование) в языке программирования PascalAB C.</p>	<p>1.Выполнение и защита практических- лабораторных работ, включая написание программ и контрольный опрос</p> <p>2.Тестирование</p>	<p>Примерные вопросы для промежуточной аттестации в форме экзамена</p>

2. Комплект оценочных средств

1. Задание для текущего контроля успеваемости

Комплект вопросов для устного и письменного опроса при защите лабораторных работ, по разделам курса

Раздел 1. Введение в программирование

Тема 1.1. Языки программирования

1. Дайте определение языку программирования. На какие две большие группы они делятся? Приведите примеры.
2. Что такое транслятор? В чем разница между компилятором и интерпретатором?
3. Назовите основные этапы решения задачи на ЭВМ.
4. Что такое алгоритм? Перечислите и охарактеризуйте основные свойства алгоритмов.
5. Каковы основные особенности и преимущества языка Паскаль? Почему он часто используется для обучения?

Тема 1.2. Типы данных в PascalABC.

1. Что такое тип данных? Для чего он нужен?
2. Перечислите основные скалярные (простые) типы данных в Pascal ABC.

3. Приведите примеры значений для каждого.
4. Чем отличается тип `integer` от `real`? В каких ситуациях используется тип
5. `longint`?
6. Что такое тип `string`? Как получить доступ к отдельному символу строки?
7. Объясните, чем константа отличается от переменной. Как они описываются в программе?

Раздел 2. Основы программирования

Тема 2.1. Операторы языка программирования PascalABC.

8. Оператор присваивания.
9. Составной оператор.
10. Условный оператор.
11. Оператор выбора.
12. Операторы цикла (с предусловием, с постусловием, с заранее известным числом повторений).
13. Одномерные массивы.
14. Двумерные массивы (свойства квадратного массива).

Раздел 3. Подпрограммы. Понятие модуля

1. Что такое подпрограмма? Какие преимущества дает использование подпрограмм?
2. В чем разница между функцией (`function`) и процедурой (`procedure`)?
3. Что такое формальные и фактические параметры? Объясните способы передачи параметров: по значению и по ссылке (`var`). Когда какой способ используется?
4. Что такое локальные и глобальные переменные? Почему не рекомендуется злоупотреблять глобальными переменными?
5. Что такое модуль? Какова его структура? Как подключить модуль к программе (`uses`)?

Раздел 4. Динамические структуры данных (Основы)

1. В чем главный недостаток массивов, который решают динамические структуры данных?
2. Что такое указатель (^)? Для чего служит оператор `new`?
3. Опишите структуру простейшего динамического односвязного списка. Из каких элементов состоит узел такого списка?
4. Какие базовые операции можно выполнять со списком? (Создание, добавление, удаление, просмотр).

Раздел 5. Объектно-ориентированное программирование (ООП) в Pascal ABC

Тема 5.1 Основные принципы объектно-ориентированного программирования (ООП)

1. Назовите три основных принципа (кита) ООП и дайте им краткое определение.
2. Что такое класс? Чем класс отличается от объекта?
3. Что такое конструктор и деструктор? Для чего они используются?
4. Что такое инкапсуляция? Как реализуются модификаторы доступа (`public`, `private`, `protected`)?

5. Что такое наследование? Приведите пример иерархии классов.

Тема 5.2 Интегрированная среда разработчика

1. Что такое интегрированная среда разработки (IDE)? Перечислите основные элементы интерфейса Pascal ABC.Net.
2. Что такое визуальное программирование? Что такое форма (Form) и компоненты (Controls)?
3. Объясните парадигму "событийно-управляемого программирования". Что такое событие, обработчик события?
4. Перечислите основные свойства компонента Button (Кнопка). Какое у него самое часто используемое событие?
5. Для чего используются компоненты TextBox (Поле ввода), Label (Надпись), ListBox (Список)?

Тема 5.3. Визуальное событийно-управляемое программирование

1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
2. Дополнительные элементы управления.
3. Свойства компонентов. Виды свойств.
4. Синтаксис определения свойств.

Тема 5.4 Разработка оконного приложения

1. Разработка функционального интерфейса приложения. Создание интерфейса приложения в PascalABC.NET при использовании встроенного дизайнера форм.
2. Разработка функциональной схемы работы приложения
3. Разработка игрового приложения.

Критерии оценки теоретических знаний:

Оценка «отлично» выставляется, если обучающийся:

- полно раскрыл содержание материала в объеме, предусмотренном программой;
- изложил материал грамотным языком в определенной логической последовательности, точно используя терминологию дисциплины и символику;
- правильно выполнил рисунки, схемы, сопутствующие ответу;
- показал умение иллюстрировать теоретические положения конкретными примерами, применяя их в новой ситуации;
- продемонстрировал усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость используемых при ответе умений и навыков;
- выполнял работу самостоятельно без помощи преподавателя.

Возможны одна - две неточности при освещении второстепенных вопросов или выкладках, которые учащийся легко исправил по замечанию преподавателя.

Оценка «хорошо» выставляется, если обучающийся:

- полно раскрыл содержание материала в объеме, предусмотренном программой;
- изложил материал грамотным языком в определенной логической последовательности, точно используя терминологию дисциплины и символику;
- правильно выполнил рисунки, схемы, сопутствующие ответу;
- показал умение иллюстрировать теоретические положения конкретными примерами, применяя их в новой ситуации;
- продемонстрировал усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость используемых при ответе умений и навыков;
- выполнял работу самостоятельно без помощи преподавателя.

Ответ при этом имеет один из недостатков:

- в изложении допущены небольшие пробелы, не исказившие содержание ответа;
- допущены один - два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя;
- допущена ошибка или более двух недочетов при освещении второстепенных вопросов или выкладках, легко исправляемые по замечанию преподавателя.

Оценка «удовлетворительно» выставляется, если:

- неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса;
- имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, схемах, выкладках, исправленные после нескольких наводящих вопросов преподавателя.

Оценка «неудовлетворительно» выставляется, если:

- не раскрыто основное содержание учебного материала;
- обнаружено незнание или непонимание учащимся большей или наиболее важной части учебного материала;
- допущены ошибки в определении понятий, при использовании терминологии, в рисунках или схемах, в выкладках, которые не исправлены после нескольких наводящих вопросов преподавателя

3. Вопросы и задание для промежуточной аттестации

Тестовые задания по темам.

1. Алфавит, идентификаторы, операции

1. Зарезервированными словами в PascalABC.net являются
1) div 2) array 3) at 4) when
5) string 6) sum 7) unit 8) while
2. Зарезервированными словами в PascalABC.net являются
1) it 2) label 3) case 4) subject
5) end 6) foreach 7) some 8) go
3. Выберите все верные утверждения. В PascalABC.net идентификатор
1) может начинаться с цифры
2) может содержать латинские буквы
3) может начинаться со знака подчеркивания
4) может содержать пробелы
5) не может иметь длину более 40 символов
4. Выберите все верные утверждения
1) Имя программы не может содержать пробелов
2) В PascalABC.net отсутствует функция tg x
3) В PascalABC.net константы можно описывать в разделе операторов
4) Обращение к функции может быть операндом в выражении

2. Стандартные функции

1. Результат действия функции odd(x) имеет тип
1) целый 2) символьный 3) логический 4) вещественный
2. К вещественным аргументам применимы функции
1) abs(x) 2) pred(x) 3) trunc(x)
4) exp(x) 5) dec(x) 6) int(x)

3. Простые типы данных. Описание переменных

1. К целому типу данных не относятся
1) shortint 2) byte 3) single 4) word
5) char 6) double 7) longword 8) decimal
2. К вещественным типам относятся
1) double 2) decimal 3) word 4) integer
5) real 6) single 7) record 8) byte
3. Выберите правильные описания констант
1) e = 2.72; 2) zero: = 0; 3) at = 'at';

4) Language: 'Pascal'; 5) no: = false; 6) yes = true;

4. Переменные объявлены правильно

- 1) var a: const; 2) var sum, zero: real;
- 3) var Language = 'Pascal'; 4) var m, con: byte;
- 5) var no = false; 6) var number: 1..9;

4. Ввод-вывод, присваивание

1. Результат действия оператора write('x=', x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

2. Результат действия оператора read(x)

- 1) ввод значения x
- 2) вывод символов 'x =' и ввод значения x
- 3) вывод значения x
- 4) вывод символов 'x=' и значения x

3. Функциями ввода данных в PascalABC.net являются

- 1) Println 2) ReadReal3 3) Read 4) WriteFormat
- 5) ReadBoolean 6) Print 7) ReadChar

4. Правильно организован ввод данных

- 1) Readln(a, b, alfa); 2) var a4 := 3.5;
- 3) var zx: integer := 2.8; 4) var (a, b): = ReadChar;
- 5) var a: = read(b); 6) var (a, b): = ReadChar2;

5.Операторы ветвления

1. Для организации алгоритма ветвления применяются операторы

- 1) repeat 2) loop 3) case
- 4) foreach 5) if 6) while

2. В результате выполнения фрагмента программы

a: = -4; b: = 1;

if a < 0 then b += a else b: = 0;

переменная b принимает значение ____

3. В результате выполнения фрагмента программы

c := 15; a := 6; d := 3;

if $a + d > c$ then $c := a * d$;

переменная c принимает значение ____

4. Правильно оформлен оператор выбора

1) var b, x: char; m, n: real;

case b of

‘@’: m := 100;

‘y’: n := 100;

‘+’: x := ‘b’;

2) var x: char; y: integer;

case x of

2: y := 5;

8: y := 2

9: y := 11;

end;

5. Определите результат выполнения операторов при вводе значения

t = 8

read(t);

case t of

1..6: y := t*t - 1;

7, 9, 13: y := 7*t - t;

10..12: y := abs(2 - t)

else y := 0

end;

6. Операторы цикла

1. В операторе цикла с предусловием тело цикла

1) обязательно выполняется более 1 раза

2) выполняется по крайней мере 1 раз

3) может не выполниться вообще

4) выполняется строго определенное число раз

2. В операторе цикла с параметром for тело цикла

1) обязательно выполняется более 1 раза

2) выполняется по крайней мере 1 раз

3) может не выполниться вообще

4) выполняется строго определенное число раз

3. Установить правильную последовательность

1 тело цикла

2 условие

3 while

4 do

4. В результате выполнения фрагмента программы

```
var y := 1; var i := 1;
```

```
while i <= 4 do
```

```
begin
```

```
y := y*i;
```

```
i += 1;
```

```
end;
```

```
write(y);
```

на экран будет выведено значение

7. Структурированные типы данных

1. Выберите все верные утверждения.

- 1) логический тип данных относится к простым
- 2) строковый тип данных относится к простым
- 3) символьный тип данных относится к структурированным
- 4) указатель относится к структурированным типам данных
- 5) перечислимый тип данных относится к простым

2. Выберите все верные утверждения

- 1) Множество не может содержать одинаковых элементов
- 2) Процедура include предназначена для проверки принадлежности элемента множеству
- 3) Сравнение строк выполняется посимвольно
- 4) Записи не могут быть элементами множества

8. Массивы

1. Выберите все правильные описания массивов

- 1) var b: array [-2..8] of integer;
- 2) var m: array [20..1] of char;
- 3) var day: array [1..'n'] of char;
- 4) var ch: array ['A'..'T'] of byte;
- 5) var letter: array ['m'..'b'] of integer;
- 6) var ord: array ['b'..'m'] of real;

2. Фрагмент кода

```
var s := a[9];  
for var i := 8 downto 1 do if a[i] < s then s := a[i];
```

вычисляет

- 1) максимальный элемент в массиве из 9 элементов
- 2) минимальный элемент в массиве из 8 элементов
- 3) максимальный элемент в массиве из 8 элементов
- 4) минимальный элемент в массиве из 9 элементов

3. Фрагмент кода

```
var s := 0; var j := 1;  
repeat  
j += 1;  
if a[j] > 0 then s += 1;  
until j > 11;
```

вычисляет

- 1) сумму положительных элементов массива из 12 элементов
- 2) количество положительных элементов в массиве из 12 элементов
- 3) сумму положительных элементов массива из 11 элементов
- 4) количество положительных элементов в массиве из 11 элементов

9. Двумерные массивы

1. Задан двумерный массив 5×5 . В результате выполнения фрагмента программы

```
for var i := 1 to 5 do  
for var j := 1 to 5 do  
a[i, j] := i*j - 7;
```

значение элемента a_{25} будет равно _____

2. Задан двумерный массив 7×7 . Следующий фрагмент программы

```
var s := 0;  
for var i := 1 to 7 do  
for var j := 1 to 7 do  
if a[i, j] > 0 then s += a[i, j];
```

определяет

- 1) сумму всех положительных элементов
- 2) модуль суммы всех отрицательных элементов
- 3) сумму обратных величин всех ненулевых элементов

4) сумму всех отрицательных элементов

10.Строки, множества

1. begin

```
var s := ReadString;
```

```
var t := ReadString;
```

```
var m := length(s);
```

```
var n := length(t);
```

```
n := abs(m — n);
```

```
writeln(n)
```

```
end.
```

Программа выводит на экран

- 1) количество символов в более длинной из двух строк
- 2) количество символов в более короткой из двух строк
- 3) число символов, на которое одна строка длиннее другой
- 4) суммарное количество символов в двух строках

2. begin

```
var s: set of integer;
```

```
s := [];
```

```
var n := ReadInteger;
```

```
loop n do
```

```
s += [Random(-30, 30)];
```

```
foreach var i in s do
```

```
if i mod 3 <> 0 then print(i)
```

```
end.
```

Код выполняет действия

- 1) задание количества элементов множества и заполнение множества числами, вводимыми с клавиатуры
- 2) заполнение массива случайными числами и вывод на экран тех, которые кратны 3
- 3) заполнение множества случайными целыми числами и вывод из них на экран некратных 3
- 4) задание количества элементов массива и вывод из них на экран некратных 3
- 5) заполнение множества случайными целыми числами и вывод на экран кратных 3

11.Файлы

1. Правильно описаны переменные файлового типа

1) var dik: file 2) var dik = file of char

3) var dik = text 4) var dik: file of byte

5) var dik: file of string 6) var dik = file of t

2. Процедура assign

1) присваивает имя создаваемому логическому файлу

2) изменяет имя объявленной ранее файловой переменной

3) открывает физический файл для записи информации

4) связывает объявленную файловую переменную с именем физического файла

12. Подпрограммы

1. Заголовок функции записан правильно

1) function ter(var a, b = real, n = integer)

2) function ter(a: real, b: real, n: integer): real

3) function ter(var a: real; b: real; n: integer): longint

4) function ter(a, b: real; n: integer): integer

2. Заголовок процедуры записан правильно

1) procedure(a, b: real; m: char): char

2) procedure sept(var a: real; const c: string)

3) procedure sept(a, c: real; n: char)

4) procedure sept(var b: real, const c = 'day')

Критерии оценки выполнения тестовых заданий

Для оценки результатов тестирования предусмотрена следующая система оценивания учебных достижений студентов:

За каждый правильный ответ ставится 1 балл, за неправильный ответ – 0 баллов.

«5» - от 91% до 100% правильных ответов

«4» - от 81% до 90% правильных ответов

«3» - от 61% до 80% правильных ответов

«2» - от 0% до 60% правильных ответов

Примерные вопросы к промежуточной аттестации.

Раздел 1. Введение в программирование

Тема 1.1. Языки программирования

Устные вопросы:

1. Дайте определение языку программирования. На какие две большие группы они делятся? Приведите примеры.
2. Что такое транслятор? В чем разница между компилятором и интерпретатором?
3. Назовите основные этапы решения задачи на ЭВМ.
4. Что такое алгоритм? Перечислите и охарактеризуйте основные свойства алгоритмов.
5. Каковы основные особенности и преимущества языка Паскаль? Почему он часто используется для обучения?

Письменное задание:

Опишите последовательность действий (алгоритм) для решения конкретной бытовой задачи (например, "Приготовление чая") в виде нумерованного списка команд. Проверьте получившийся алгоритм на свойство массовости и определенности.

Тема 1.2. Типы данных

Устные вопросы:

2. Что такое тип данных? Для чего он нужен?
3. Перечислите основные скалярные (простые) типы данных в Pascal ABC. Приведите примеры значений для каждого.
4. Чем отличается тип integer от real? В каких ситуациях используется тип longint?
5. Что такое тип string? Как получить доступ к отдельному символу строки?
6. Объясните, чем константа отличается от переменной. Как они описываются в программе?

Письменное задание:

1. Даны описания переменных. Найдите ошибки, если они есть, и исправьте их:

```
``pascal
var
  a, b: Integer;
  summa: Real;
  flag: Bool;
  name: String[50];
  sym: Char;
  'num': Integer; // Ошибка здесь
...
```

2. Напишите фрагмент программы, который описывает переменные для хранения: количества студентов в группе (целое), средней оценки (вещественное), признака сдачи экзамена (логическое), фамилии студента (строка).

Раздел 2. Основы программирования

Тема 2.1. Операторы языка программирования

Устные вопросы:

1. Перечислите и кратко охарактеризуйте три основных структуры алгоритмов (следование, ветвление, цикл).
2. Объясните разницу между операторами if...then и if...then...else. Приведите примеры.
3. В чем разница между циклами с предусловием (while), постусловием (repeat...until) и параметрическим циклом (for)? Когда какой цикл предпочтительнее?
4. Что такое составной оператор и для чего он используется?
5. Как работает оператор выбора case? Какие типы данных можно использовать в селекторе?

Письменное задание:

1. Что выведет на экран данная программа? Пошагово объясните результат.

```
``pascal
var x, y: integer;
begin
  x := 10;
  y := 5;
  if x > y then
    x := x - y
  else
    y := y - x;
  writeln('x=', x, ' y=', y);
end.
``
```

2. Написать программу вычисления суммы всех четных чисел в диапазоне от 130 до 256 включительно.
3. Написать программу поиска суммы всех трехзначных чисел, кратных 3, но не кратных 7.
4. Создать базу данных магазина, содержащую поля: наименование, цена за единицу, количество и стоимость. Поле Стоимость вычисляемое (цена на количество).

5. Напишите программу сортировки одномерного массива из 20 элементов по методу «пузырька» (по убыванию).
6. Напишите программу поиска максимального элемента одномерного массива из 20 элементов.
7. Напишите программу поиска минимального элемента одномерного массива из 20 элементов.
8. Напишите программу поиска первого отрицательного элемента одномерного массива из 20 элементов.
9. Напишите программу поиска последнего положительного элемента одномерного массива из 20 элементов.
10. Напишите фрагмент программы, который описывает переменные для хранения: количества студентов в группе (целое), средней оценки (вещественное), признака сдачи экзамена (логическое), фамилии студента (строка).

Раздел 3. Подпрограммы. Понятие модуля

Тема 3.1. Процедуры и функции

Устные вопросы:

1. Что такое подпрограмма? Какие преимущества дает использование подпрограмм?
2. В чем разница между функцией (function) и процедурой (procedure)?
3. Что такое формальные и фактические параметры? Объясните способы передачи параметров: по значению и по ссылке (var). Когда какой способ используется?
4. Что такое локальные и глобальные переменные? Почему не рекомендуется злоупотреблять глобальными переменными?
5. Понятие рекурсии. Рекурсивные алгоритмы.
6. Что такое модуль? Какова его структура? Как подключить модуль к программе (uses)?

Письменное задание:

1. Напишите программу вычисления факториала, используя рекурсивную функцию.
2. Напишите программу вычисления суммы нечетных элементов двумерного массива, размерностью 5x5, используя процедуры с параметром.
3. Напишите программу вычисления количества двузначных элементов двумерного массива, размерностью 5x5, используя процедуры с параметром.
4. Напишите программу вычисления суммы ряда $r = 1 + 2 + 3 + \dots + N$, $N = 25$ используя процедуры с параметром.
5. Объясните, будет ли меняться исходная переменная `a` после вызова процедуры в данном коде:

```
``pascal
procedure Test(x: integer; var y: integer);
```

```

begin
  x := x * 2;
  y := y + 10;
end;

var a, b: integer;
begin
  a := 5; b := 5;
  Test(a, b);
  writeln(a, ' ', b); // Что будет выведено?
end.
```

```

### Тема 3.2. Структуризация в программировании

Устные вопросы:

1. Основы структурного программирования.
2. Методы структурного программирования.

### Тема 3.3. Модульное программирование

Устные вопросы:

1. Модульное программирование.
2. Понятие модуля. Структура модуля.
3. Компиляция и компоновка программы.
4. Стандартные модули.

### Раздел 4. Динамические структуры данных (Основы)

Устные вопросы:

1. В чем главный недостаток массивов, который решают динамические структуры данных?
2. Что такое указатель (^)? Для чего служит оператор new?
3. Опишите структуру простейшего динамического односвязного списка. Из каких элементов состоит узел такого списка?
4. Какие базовые операции можно выполнять со списком? (Создание, добавление, удаление, просмотр).

Письменное задание:

1. Опишите тип записи (record) для представления элемента списка, хранящего информацию о книге: автор (строка), название (строка), год издания (целое).
2. Нарисуйте схему (блок-схему) односвязного списка из трех элементов.

### Раздел 5. Объектно-ориентированное программирование (ООП) в Pascal ABC

Устные вопросы:

1. Назовите три основных принципа (кита) ООП и дайте им краткое определение.
2. Что такое класс? Чем класс отличается от объекта?

3. Что такое конструктор и деструктор? Для чего они используются?
4. Что такое инкапсуляция? Как реализуются модификаторы доступа (public, private, protected)?
5. Что такое наследование? Приведите пример иерархии классов.

Письменное задание:

1. Объявите класс TRectangle (Прямоугольник) с приватными полями width и height (ширина, высота). Реализуйте: Публичные методы для установки значений (SetSize) и получения площади (GetArea).

Конструктор с параметрами.

2. Объявите класс TStudent, наследующийся от базового класса THuman. Какие новые поля и методы можно добавить в класс-потомок?

Событийно-управляемая модель программирования.

Компонентно-ориентированный подход.

#### Тема 5.2 Среда разработки и событийное программирование (Pascal ABC Visual)

Устные вопросы:

1. Что такое интегрированная среда разработки (IDE)? Перечислите основные элементы интерфейса Pascal ABC.Net.
2. Что такое визуальное программирование? Что такое форма (Form) и компоненты (Controls)?
3. Объясните парадигму "событийно-управляемого программирования". Что такое событие, обработчик события?
4. 4. Перечислите основные свойства компонента Button (Кнопка). Какое у него самое часто используемое событие?
5. Для чего используются компоненты TextBox (Поле ввода), Label (Надпись), ListBox (Список)?

#### Тема 5.3. Визуальное событийно-управляемое программирование

Устные вопросы:

6. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.
7. Дополнительные элементы управления.
8. Свойства компонентов. Виды свойств.
9. Синтаксис определения свойств.
10. Назначения свойств и их влияние на результат.
11. Управление объектом через свойства.
12. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.

#### Тема 5.4 Разработка оконного приложения

Письменное задание:

1. Разработка интерфейса приложения для ввода одномерного массива с клавиатуры и вывода на экран максимального четного элемента.
2. Разработка интерфейса приложения для ввода одномерного массива с клавиатуры и вывода на экран максимального нечетного элемента.
3. Разработка интерфейса приложения для ввода двумерного массива, размерностью 4x4. И вывода на экран суммы его элементов.
4. Разработка интерфейса приложения для ввода двумерного массива, размерностью 4x4. И вывода на экран произведения его элементов, стоящих на главной диагонали.
5. Разработка интерфейса приложения для ввода двумерного массива, размерностью 4x4. И вывода на экран среднего арифметического его элементов, стоящих на побочной диагонали.
6. Разработка интерфейса приложения расчета площади прямоугольного треугольника, по заданным катетам.
7. Разработка интерфейса приложения расчета площади трапеции, по заданным основаниям и высоте.
8. Разработка интерфейса приложения расчета площади окружности с заданным радиусом.
9. Разработка интерфейса приложения расчета объема конуса по заданным радиусу и высоте.

#### Тема 5.5 Этапы разработки приложений

Устные вопросы:

1. Этапы разработки приложений

#### Тема 5.6 Иерархия классов.

Устные вопросы:

1. Классы ООП: виды, назначение, свойства, методы, события.
2. Перегрузка методов

#### 3.Критерии оценки

##### **Критерии оценки результатов учебной дисциплины:**

**«отлично»** ставится за работу, выполненную без ошибок и недочетов или имеющую не более одного недочета.

**«хорошо»** ставится за работу, выполненную полностью, но при наличии в ней не более одной негрубой ошибки и одного недочета или не более двух недочетов.

**«удовлетворительно»** ставится за работу, выполненную не менее чем на  $\frac{1}{2}$  или допустил: Не более двух грубых ошибок