

ПРОГРАММА
вступительного испытания
«Информационные технологии и программирование»

Составлена на основе «Федерального государственного образовательного стандарта среднего профессионального образования», «Кодификатора элементов содержания и требований к уровню подготовки выпускников образовательных организаций для проведения вступительного экзамена по основам программирования».

В соответствии с этими требованиями абитуриент должен:

уметь:

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.

знать:

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования.

Раздел 1. Перечень элементов содержания, проверяемых на вступительном экзамене по информационным технологиям

Раздел 1. Основные принципы алгоритмизации

1.1. Понятие и свойства алгоритма.

1.1.1. Понятие алгоритма, его свойства

1.1.2. . Сущность алгоритмизации вычислительного процесса

1.1.3. Способы описания алгоритмов

1.2. Описание алгоритма в виде блок-схем

1.2.1. Структурный подход к разработке блок-схем.

1.2.2. Базовые алгоритмические структуры: линейный, условный, циклический.

1.2.3. Приемы использования базовых алгоритмических структур

1.3. Данные в алгоритмическом процессе. Типы данных

1.3.1. Основные базовые типы данных и их характеристика.

1.3.2. Структурированные типы данных и их характеристика

1.4. Массивы

1.4.1 Понятие массива.

1.4.2 Размерность массива.

1.4.3. Обращение к элементам массива.

1.4.4. Инициализация массива.

1.4.5. Алгоритмы формирования массива

1.5. Массивы

1.5.1. Поиск максимального элемента в одномерном массиве.

1.5.2. Сортировка одномерного массива.

1.5.3. Определение элементов массива, обладающих заданными свойствами

1.6. Подчиненные алгоритмы

1.6.1. Понятие подалгоритма.

1.6.2. Модульный подход к разработке алгоритмов.

1.6.3. Правила выделения и оформления подалгоритма.

1.6.4. Функции и процедуры.

1.6.5. Выполнение подалгоритма.

1.6.6. Формальные и фактические параметры.

1.6.7. Примеры разработки подалгоритмов

Раздел 2.

Средства и методы программирования

2.1. Языки и системы программирования

- 2.1.1. Эволюция языков программирования.
- 2.1.2. Классификация языков программирования.
- 2.1.3. Элементы языков программирования.
- 2.1.4. Понятие системы программирования.
- 2.1.5. Исходный, объектный и загрузочный модули.
- 2.1.6. Интегрированная среда программирования

2.2. Методы программирования

- 2.2.1. Структурный, модульный, объектно-ориентированный методы программирования.
- 2.2.2. Достоинства и недостатки методов программирования

2.3. Общие принципы разработки программ

- 2.3.1. Общие принципы разработки программного обеспечения.
- 2.3.2. Жизненный цикл программного обеспечения.
- 2.3.3. Этапы решения задачи на компьютере

Раздел 3.

Программирование на языке Turbo Pascal

3.1. Особенности языка Turbo Pascal

- 3.1.1. Назначение языка.
- 3.1.2. Состав языка.
- 3.1.3. Алфавит языка.
- 3.1.4. Лексемы языка.
- 3.1.5 Структура программы на языке.

3.2. Типы данных языка Turbo Pascal

- 3.2.1. Концепция типов данных.
- 3.2.2. Стандартные типы данных.
- 3.2.3. Переменные и константы, объявление переменных и констант.
- 3.2.4. Инициализация переменных

3.3. Операции и выражения языка Turbo Pascal

- 3.3.1. Понятия выражения, операции, операторов и операндов.
- 3.3.2. Операции языка Turbo Pascal и их приоритет
- 3.3.3. Блочный оператор.
- 3.3.4. Составной оператор

3.4. Среда программирования Turbo Pascal

- 3.4.1. Особенности интегрированной среды.
- 3.4.2. Режимы работы в интегрированной среде.
- 3.4.3. Компоненты интегрированной среды.
- 3.4.4. Оформление и отладка программ.
- 3.4.5. Встроенный текстовый редактор.
- 3.4.6. Выполнение программы.
- 3.4.7. Директивы компилятора и включение их в текст программы.
- 3.4.8. Виды программных ошибок

3.5. Ввод-вывод данных в языке Turbo Pascal

- 3.5.1. Подходы к вводу-выводу данных.
- 3.5.2. Форматированный ввод-вывод с помощью функций read, readln, write, writeln.
- 3.5.3. Управление символьным выводом на экран

3.6. Программирование условных алгоритмов на Turbo Pascal

- 3.6.1. Операторы полного и неполного ветвления (if-then и if-then-else).
- 3.6.2. Вложенные условные операторы.
- 3.6.3. Оператор выбора case.
- 3.6.4. Оператор безусловного перехода. Метки.
- 3.6.5. Примеры разработки программ по условным алгоритмам

3.7. Программирование циклических алгоритмов на Turbo Pascal

- 3.7.1. Оператор цикла с предусловием while.
- 3.7.2. Оператор цикла с постусловием repeat.
- 3.7.3. Оператор цикла с параметром for.
- 3.7.4. Примеры разработки программ по циклическим алгоритмам
- 3.8. Массивы в языке Turbo Pascal**
- 3.8.1. Массивы в Turbo Pascal.
- 3.8.2. Обращение к элементам массивов.
- 3.8.3. Объявление массивов.
- 3.8.4. Использование пользовательского типа при объявлении массива.
- 3.8.5. Инициализация массивов.
- 3.8.6. Примеры программ обработки массивов
- 3.9. Строковый тип данных в Turbo Pascal**
- 3.9.1. Строки.
- 3.9.2. Объявление строк.
- 3.9.3. Строковые константы.
- 3.9.4. Инициализация строк
- 3.10. Обработка строковых данных в Turbo Pascal**
- 3.10.1. Функции работы со строками. Примеры программ
- 3.11. Множественный тип данных в Turbo Pascal**
- 3.11.1. Понятие множества.
- 3.11.2. Конструктор множества.
- 3.11.3. Объявление множеств.
- 3.11.4. Свойства множеств.
- 3.11.5. Операции над множествами
- 3.12. Записи в Turbo Pascal**
- 3.12.1. Понятие записи.
- 3.12.2. Структура записи.
- 3.12.3. Поля записи.
- 3.12.4. Объявление записи.
- 3.12.5. Обращение к полям записи.
- 3.12.6. Примеры программ
- 3.13. Файлы и файловые переменные в Turbo Pascal**
- 3.13.1. Понятие файла.
- 3.13.2. Типы файлов.
- 3.13.3. Файловые переменные
- 3.14. Функции работы с файлами в Turbo Pascal**
- 3.14.1. Запись в файл.
- 3.14.2. Чтение из файла.
- 3.14.3. Прямой доступ к записям файла.
- 3.14.4. Примеры программ
- 3.15. Указатели в Turbo Pascal**
- 3.15.1. Статические и динамические данные.
- 3.15.2. Понятие указателя.
- 3.15.3. Объявление указателей.
- 3.15.4. Управление выделением и освобождением динамической памяти на логическом и физическом уровнях
- 3.16. Списки в Turbo Pascal**
- 3.16.1. Понятие списка.
- 3.16.2. Создание и обработка списков.
- 3.16.3. Примеры программ
- 3.17. Графические режимы работы монитора в Turbo Pascal**
- 3.17.1. Установка графического режима работы монитора.
- 3.17.2. Цвет фона и рисунка.
- 3.17.3. Графические координаты.
- 3.17.4. Графическое окно
- 3.18. Создание графических изображений в Turbo Pascal**
- 3.18.1. Графические примитивы.

- 3.18.2. Закраски и заполнения.
- 3.18.3. Построение графиков функций.
- 3.18.4. Примеры программ
- 3.19. Процедуры пользователя в Turbo Pascal**
- 3.19.1. Процедуры.
- 3.19.2. Описание процедуры.
- 3.19.3. Оператор вызова процедуры
- 3.20. Функции пользователя в Turbo Pascal**
- 3.20.1. Функция.
- 3.20.2. Описание функции.
- 3.20.3. Оператор вызова функции.
- 3.20.4. Рекурсивные функции
- 3.21. Модульное программирование в Turbo Pascal**
- 3.21.1. Понятие модуля.
- 3.21.2. Описание модуля.
- 3.21.3. Интерфейсная, исполнимая и инициализирующие части
- 3.21.4. Создание и использование модулей

Раздел 4.

Объектно-ориентированное программирование в Turbo Pascal

4.1. Основные понятия объектно-ориентированного программирования

- 4.1.1. История развития объектно-ориентированного программирования
- 4.1.2. Основные понятия объектно-ориентированного программирования: объекты и классы.
- 4.1.3. Основные принципы объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм

4.2. Классы в Turbo Pascal

- 4.2.1. Определение класса.
- 4.2.2. Обращение к полям и методам класса.
- 4.2.3. Создание объектов (экземпляров класса).
- 4.2.4. Инициализация полей объектов

4.3. Наследование в Turbo Pascal

- 4.3.1. Механизм наследования.
- 4.3.2. Иерархия классов.
- 4.3.3. Присваивание объектов

4.4. Полиморфизм в Turbo Pascal

- 4.4.1. Механизм простого полиморфизма.
- 4.4.2. Применение простого полиморфизма.
- 4.4.3. Механизм сложного полиморфизма.
- 4.4.4. Использование процедур с полиморфным объектом

4.5. Динамические объекты

- 4.5.1. Объявление динамического объекта.
- 4.5.2. Контроль распределения памяти.
- 4.5.3. Пример организации динамического объекта с динамическими полями.
- 4.5.4. Создание статического объекта с динамическими полями

4.6. Разработка программы с использованием ООП

- 4.6.1. Этапы разработки: анализ и объектная декомпозиция предметной области задачи, логическое проектирование, физическое проектирование, создание библиотеки классов и основной программы.
- 4.6.2. Примеры программы

Раздел 2. Перечень требований к уровню подготовки выпускников, достижение которого проверяется на вступительном экзамене по информационным технологиям
Знать/понимать/уметь:

1. Работать в среде программирования;

2. Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.
3. Этапы решения задачи на компьютере;
4. Типы данных;
5. Базовые конструкции изучаемых языков программирования;
6. Принципы структурного и модульного программирования;
7. принципы объектно-ориентированного программирования.

Основная литература

- 1) Абрамов В.Г., Трифонов Н.П., Трифонова Г.Н. Введение в язык Pascal. Учебное пособие. – М.: Кнорус, 2011. – 384 с.
- 2) Голицина О. Л., Попов И. И. Основы алгоритмизации и программирования: Учебное пособие. – М.: ФОРУМ, 2009. – 1168 с.
- 3) Культин Н. Самоучитель программирования в Turbo Pascal 7.0 и Delphi. СПб.: BHV, 2009. – 407 с.
- 4) Микрюков В.Ю. Алгоритмизация и программирование.- М.: Феникс, 2009. – 304 с.
- 5) Сальников Ю.Н. Программирование. Базовый курс. – М.: Маркет ДС, 2011. – 336 с.
- 6) Семакин И.Г., Шестаков А.П. Основы алгоритмизации и программирования: Учебник. – М.: Академия, 2010. – 400 с.
- 7) Сухарев М. Turbo Pascal 7.0. Теория и практика программирования. – М.: Наука и техника, 2009. – 544 с.

Дополнительная литература

- 1) Давыдова Н.А., Боровская Е.В. Программирование. Учебное пособие. – М.: БИНОМ, 2009. – 238 с.
- 2) Зеленьяк О.П. Практикум программирования на Turbo Pascal. – М.: Книга по требованию, 2010. – 896 с.
- 3) Зеленьяк О.П. Современный задачник по Турбо Паскалю. – М.: ДМК-Пресс, 2010. – 320 с.
- 4) Канцедаль С.А. Алгоритмизация и программирование. – М.: Форум, 2010. – 352 с.
- 5) Карпиленко Е.В. Основы программирования. – М.: Феникс, 2009. – 318 с.
- 6) Культин Н. Turbo Pascal в задачах и примерах. – СПб.: BHV, 2010. – 256 с.
- 7) Потопахин В. Искусство алгоритмизации. – М.: ДМК-Пресс, 2011. – 320 с.
- 8) Светозарова Г.И., Сигитов Е.В., Козловский А.В. Практикум по программированию на алгоритмических языках. М.: Книга по требованию, 2012. – 432 с.
- 9) Фаронов В.В. Turbo Pascal 7.0. Практика программирования. – М.: Кнорус, 2012. – 414 с.

Интернет-ресурсы:

- 1) Федеральный центр информационно-образовательных ресурсов [Официальный сайт]. URL: <http://fcior.edu.ru/> (дата обращения: 2.08.2013).
- 2) Учебный курс – Инструменты, алгоритмы и структуры данных [Электронный ресурс] // Интернет-Университет Информационных технологий – дистанционное образование [Официальный сайт]. URL: <http://www.intuit.ru/departments/se/ialgdate/> (дата обращения: 2.08.2013).
- 3) Учебный курс – Программирование на языке Pascal [Электронный ресурс] // Интернет-Университет Информационных технологий – дистанционное образование [Официальный сайт]. URL: <http://www.intuit.ru/departments/pl/plpascal/> (дата обращения: 2.08.2013).
- 4) Учебный курс – Программирование на языке высокого уровня Паскаль [Электронный ресурс] // Интернет-Университет Информационных технологий – дистанционное образование [Официальный сайт]. URL: <http://www.intuit.ru/departments/pl/cpp/> (дата обращения: 2.08.2013).
- Учебный курс – Введение в программирование на Delphi [Электронный ресурс] // Интернет-Университет Информационных технологий – дистанционное образование [Официальный сайт]. URL: <http://www.intuit.ru/departments/pl/intdelphi/> (дата обращения: 2.08.2013).

Варианты тестовых заданий по вступительному испытанию
«Информационные технологии и программирование»
для формирования вступительных экзаменационных материалов для лиц,
имеющих среднее профессиональное образование при приеме на обучение по
программам бакалавриата/специалитета.

1. Алгоритм – это:
2. Суть такого свойства алгоритма, как результативность, заключается в том, что:
3. Суть такого свойства алгоритма, как формальность, заключается в том, что:
4. Какое свойство алгоритма выражает утверждение: Исполнение алгоритма с одними и теми же исходными данными всегда должно приводить к одному и тому же результату.
5. Переменная в программировании наиболее полно характеризуется:
6. Какое значение будет результатом вычисления выражения $6 + 37 \div 6$?
7. Какая из указанных функций возвращает символ по его коду?
8. Какая из указанных функций преобразует величину вещественного типа в целый тип?
9. Какое значение будет результатом вычисления выражения $\text{sqr}(2*5) - 14 \bmod (2*7) - 40$?
10. Uses – это раздел:
11. Какая из процедур на пользовательском экране создаёт прямоугольную область ввода/вывода информации?
12. Какое действие производит процедура ClrScr?
13. Какого типа будет результат выражения: $15 \bmod 3 + 8$?
14. Чему равно значение переменной K после выполнения следующих действий:
 $K:=10; S:=K+5; K:=S*2; S:=K-10$?
15. При каком значении переменной X, переменная Y будет принимать значение 36 после выполнения программной строки: $Y:=\text{SQR}(X) + (X \bmod 3)$; ?
16. Какая из процедур обеспечивает ввод информации с помощью клавиатуры?
17. Для чего предназначена процедура Write(A, D);?
18. Что будет выведено на экран после выполнения: $\text{Write}('15+15', '=', 15+15)$?
19. На кого рассчитан алгоритм, написанный на естественном языке?
20. Суть такого свойства алгоритма, как массовость, заключается в том, что:
21. Суть такого свойства алгоритма, как точность, заключается в том, что:
22. Какое свойство алгоритма выражает утверждение: На каждом шаге алгоритма у исполнителя должно быть достаточно информации, чтобы его выполнить.
23. Константа в программировании наиболее полно характеризуется:
24. Какое значение будет результатом вычисления выражения $6+37 \bmod 6$?
25. Какая из указанных процедур увеличивает величину аргумента на 1?