

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И  
ПРОГРАММИРОВАНИЕ**

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль): «Электронный бизнес»

Форма обучения: очная, заочная

Квалификация (степень) выпускника: бакалавр

Срок обучения: очная форма – 4 года, заочная форма – 4 года 6 мес.

Вид учебной работы	Трудоемкость, часы (з.е.)	
	Очная форма	Заочная форма
1. Контактная работа обучающихся с преподавателем	36(1)	8(0,22)
1. Аудиторные занятия, часов всего, в том числе:	34(0,94)	6(0,17)
• лекции	16(0,44)	2(0,06)
• лабораторные	18(0,5)	4(0,11)
Промежуточная аттестация (контактная работа)	2(0,06)	2(0,06)
2. Самостоятельная работа студентов, всего	56(1,56)	129(3,58)
• др. формы самостоятельной работы	56(1,56)	129(3,58)
3. Промежуточная аттестация: экзамен	52(1,44)	7(0,19)
Итого	144(4)	144(4)

Ахмедова А.М. Объектно-ориентированный анализ и программирование: Рабочая программа дисциплины (модуля). Казань: Казанский кооперативный институт (филиал) Российского университета кооперации, 2019 г. – 54 с.

Рабочая программа по дисциплине (модулю) «Объектно-ориентированный анализ и программирование» по направлению подготовки 38.03.05 Бизнес-информатика, направленность «Электронный бизнес» составлена Ахмедовой А.М., доцентом кафедры естественных дисциплин, сервиса и туризма Казанского кооперативного института (филиала) Российского университета кооперации в соответствии с требованиями Государственного образовательного стандарта высшего образования по специальности «Бизнес-информатика», утвержденного приказом Министерства образования и науки Российской Федерации от «11» августа 2016 г. № 1002, и учебными планами по направлению подготовки 38.03.05 Бизнес-информатика, направленность (профиль) «Электронный бизнес» (год начала подготовки -2019).

Рабочая программа:

**обсуждена и рекомендована** к утверждению решением кафедры естественных дисциплин, сервиса и туризма Казанского кооперативного института (филиала) Российского университета кооперации от «13» марта 2019 г., протокол № 7.

**одобрена** Научно-методическим советом Казанского кооперативного института (филиала) от 03.04.2019, протокол № 5.

утверждена Ученым советом Российского университета кооперации от 18.04.2019, протокол № 4.

© АНОО ВО ЦС РФ  
«Российский университет  
кооперации, Казанский  
кооперативный институт  
(филиал), 2019  
© Ахмедова А.М., 2019

## СОДЕРЖАНИЕ

1. Цели и задачи освоения дисциплины (модуля).....	4
2. Место дисциплины (модуля) в структуре образовательной программы .....	4
3. Перечень планируемых результатов обучения по дисциплине (модулю).....	4
4. Объем дисциплины (модуля) и виды учебной работы .....	5
5. Содержание дисциплины (модуля).....	6
5.1. Содержание разделов, тем дисциплины (модуля).....	6
5.2. Междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами (модулями) .....	7
5.3. Разделы, темы дисциплины (модуля) и виды занятий.....	7
6. Лабораторный практикум .....	7
7. Практические занятия (семинары).....	8
8. Примерная тематика курсовых проектов (работ).....	8
9. Самостоятельная работа студента .....	9
10. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).....	9
11. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля).....	10
12. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля) .....	10
13. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости).....	11
14. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).....	11
15. Методические указания для обучающихся по освоению дисциплины (модуля) .....	12
16. Методические рекомендации по организации изучения дисциплины для преподавателей, образовательные технологии.....	13
<b>ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) .....</b>	<b>14</b>
1. Паспорт фонда оценочных средств .....	15
1.1. Компетенции, формируемые в процессе изучения дисциплины.....	15
1.2. Сведения об иных дисциплинах (преподаваемых, в том числе, на других кафедрах) участвующих в формировании данных компетенций .....	15
1.3. Этапы формирования и программа оценивания контролируемой компетенции.....	15
1.4. Показатели и критерии оценивания компетенций, шкала .....	17
2. Типовые контрольные задания для оценки результатов обучения по дисциплине и иные материалы для подготовки к промежуточной аттестации .....	20
2.1. Материалы для подготовки к промежуточной аттестации .....	20
2.2. Комплект экзаменационных билетов для проведения промежуточной аттестации.....	27
Комплект тестовых заданий для проведения экзамена по дисциплине.....	28
2.3. Критерии оценки для проведения экзамена по дисциплине.....	30
2.4. Методические материалы, определяющие процедуру оценивания по дисциплине.....	31
<b>ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ТЕКУЩЕЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) .....</b>	<b>32</b>
Вопросы для проведения опроса.....	33
Тестовые вопросы.....	35
Комплект заданий для проведения текущей аттестации №1 .....	40
Комплект заданий для проведения текущей аттестации №2 .....	48

## **1. Цели и задачи освоения дисциплины (модуля)**

Целью освоения дисциплины «Объектно-ориентированный анализ и программирование» является формирование у студентов теоретических знаний и практических навыков по разработке программного обеспечения (ПО) для решения экономических и расчетных задач с применением современных методов и технологий программирования.

Задачи дисциплины:

- изучение принципов объектно-ориентированного подхода к разработке ПО;
- приобретение навыков визуальной разработки приложений для решения экономических задач;
- овладение опытом создания программ с развитым графическим интерфейсом.

## **2. Место дисциплины (модуля) в структуре образовательной программы**

Дисциплина относится к базовой части дисциплин блока Б1 «Дисциплины (модули)»

Для изучения дисциплины необходимы следующие знания, умения и владения навыками, формируемые предшествующими дисциплинами:

Теоретические основы информатики (ОПК-3)

Программирование (ОПК-3; ПК-18)

## **3. Перечень планируемых результатов обучения по дисциплине (модулю)**

Изучение учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

ОПК-3 - способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях;

ПК-18 - способность использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования.

В результате изучения дисциплины обучающийся должен:

Формируемые компетенции (код компетенции)	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций	Наименование оценочного средства
ОПК-3 ПК-18	Знать основные методы анализа и технологии программирования.	<i>Опрос Тест</i>
	Знать основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций	
	Уметь разрабатывать алгоритмы решения; программировать задачи обработки данных в предметной области; выполнять тестирование и отладку программ.	<i>Лабораторные работы</i>
	Умеет применять полученные в процессе обучения знания при решении задач профессиональной деятельности;	
	Владеть методами и инструментальными средствами разработки программ.	<i>Контрольные работы</i>
	Владеть информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;	

#### 4. Объем дисциплины (модуля) и виды учебной работы

##### *очная форма обучения*

Вид учебной деятельности	Часов	
	Всего	По семестрам
1. Контактная работа обучающихся с преподавателем:	36	36
Аудиторные занятия всего, в том числе:	34	34
Лекции	16	16
Лабораторные работы	18	18
Промежуточная аттестация (контактная работа)	2	2
2. Самостоятельная работа студента всего, в том числе:	56	56
Другие виды самостоятельной работы	56	56
Вид промежуточной аттестации: экзамен	52	52
ИТОГО:	часов	144
Общая трудоемкость	зач. ед.	4

##### *заочная форма обучения*

Вид учебной деятельности	Часов	
	Всего	За курс
1. Контактная работа обучающихся с преподавателем:	8	8
Аудиторные занятия всего, в том числе:	6	6
Лекции	2	2

Вид учебной деятельности	Часов	
	Всего	За курс
		2
Лабораторные работы	4	4
Промежуточная аттестация (контактная работа)	2	2
2. Самостоятельная работа студента всего, в том числе:	129	129
Другие виды самостоятельной работы	129	129
Вид промежуточной аттестации: экзамен	7	7
ИТОГО:	часов	144
Общая трудоемкость	зач. ед.	4

## 5. Содержание дисциплины (модуля)

### 5.1. Содержание разделов, тем дисциплины (модуля)

#### Тема 1. Объектно-ориентированный подход к проектированию и разработке программ

Интегрированная среда разработки (ИСР) – структура, возможности. Проект, файлы, входящие в состав проекта. Программирование под Windows. Библиотека функций Windows API. Основные принципы объектно-ориентированного программирования. Особенности языка программирования Turbo Delphi (Visual Basic). Классы и объекты, поля, свойства, методы, события. Конструкторы и деструкторы.

#### Тема 2. Визуальное программирование

Форма: свойства и методы формы. События, организация реакции на них. Визуальные компоненты, обзор, использование библиотек. Компоненты ввода и отображения текстовой информации. Компоненты – элементы управления. Компоненты - меню. Компоненты внешнего оформления. Организация диалогов. Компоненты – диалоги.

#### Тема 3. Разработка графического интерфейса пользователя

Требования к интерфейсу, многооконные приложения. События клавиатуры и "мыши". Технология Drag&Drop. "Продвинутые" компоненты для организации интерфейса пользователя.

#### Тема 4. Графика и мультимедиа

Компоненты отображения графической информации. Канва, перо, кисть, их свойства и методы. Мультимедиа и анимация.

#### Тема 5. Развитые элементы приложений

Помощь пользователю: Help-система, подсказки, строка состояния. Разработка и добавление компонентов. Сообщения Windows и их обработка.

COM-технология программирования. Свойства, методы серверов MS Word и MS Excel.

## 5.2. Междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами (модулями)

Дисциплина «Объектно-ориентированный анализ и программирование» формирует ОПК-3, ПК-18 компетенции, необходимые в дальнейшем для формирования компетенций ПК-6, ПК-13, ПК-19.

## 5.3. Разделы, темы дисциплины (модуля) и виды занятий

### *очная форма обучения*

№ п/п	Наименование темы дисциплины (модуля)	Виды занятий, включая самостоятельную работу студентов (в часах)			
		Лекции	Лабораторные занятия	Самостоятельная работа	Всего
1.	Объектно-ориентированный подход к проектированию и разработке программ	2	2	10	14
2.	Визуальное программирование	4	6	10	20
3.	Разработка графического интерфейса пользователя	4	4	10	18
4.	Графика и мультимедиа	4	4	12	20
5.	Развитые элементы приложений	2	2	14	18
	<b>Итого</b>	<b>16</b>	<b>18</b>	<b>56</b>	<b>90</b>

### *заочная форма обучения*

№ п/п	Наименование темы дисциплины (модуля)	Виды занятий, включая самостоятельную работу студентов (в часах)			
		Лекции	Лабораторные занятия	Самостоятельная работа	Всего
1.	Объектно-ориентированный подход к проектированию и разработке программ	-	1	24	25
2.	Визуальное программирование	-	3	24	27
3.	Разработка графического интерфейса пользователя	1	-	24	25
4.	Графика и мультимедиа	1	-	27	28
5.	Развитые элементы приложений	-	-	30	30
	<b>Итого</b>	<b>2</b>	<b>4</b>	<b>129</b>	<b>135</b>

## 6. Лабораторный практикум

Лабораторные занятия проводятся с целью формирования компетенций обучающихся, закрепления на практике полученных на лекциях и в процессе самостоятельного изучения обучающимися специальной литературы.

**очная форма обучения**

№ п/п	Наименование раздела, темы учебной дисциплины (модуля)	Тематика лабораторных занятий	Трудоёмкость (час.)
1	Объектно-ориентированный подход к проектированию и разработке программ	Создание консольного приложения Разработка класса, переопределение операций Разработка системы классов	2
2	Визуальное программирование	Использование компонентов вкладки Standard	2
		Использование компонентов меню	2
		Использование компонентов кладки Additional	
3	Разработка графического интерфейса пользователя	Использование диалоговых компонентов	2
		Многооконные приложения. Обработка событий клавиатуры и "мыши"	2
4	Графика и мультимедиа	Технология Drag&Drop	2
		Построение графических изображений.	2
5	Развитые элементы приложений	Мультимедиа и анимация	2
		Разработка Help-системы Разработка и добавление компонентов	2
<b>Итого</b>			<b>18</b>

**заочная форма обучения**

№ п/п	Наименование раздела, темы учебной дисциплины (модуля)	Тематика лабораторных занятий	Трудоёмкость (час.)
1	Объектно-ориентированный подход к проектированию и разработке программ	Создание консольного приложения Разработка класса, переопределение операций Разработка системы классов	1
2	Визуальное программирование	Использование компонентов вкладки Standard	1
		Использование компонентов меню	1
		Использование компонентов кладки Additional	
Итого		Использование диалоговых компонентов	1
		<b>4</b>	

**7. Практические занятия (семинары)**

Практические занятия не предусмотрены учебными планами.

**8. Примерная тематика курсовых проектов (работ)**

Курсовые проекты (работы) не предусмотрены учебными планами.

## 9. Самостоятельная работа студента

№ п/п	Наименование раздела, темы учебной дисциплины (модуля)	Виды самостоятельной работы студента	Оценочное средство
1.	Объектно-ориентированный подход к проектированию и разработке программ Рекомендации: обратить внимание на основные понятия: инкапсуляция, наследование, полиморфизм, класс, объект.	Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам: 1. Создание консольного приложения. Разработка класса, переопределение операций. Разработка системы классов	Устный опрос
2.	Визуальное программирование Рекомендации: обратить внимание на преимущества и этапы визуального проектирования.	Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам: 1. Использование компонентов вкладки Standard Использование компонентов меню 2. Использование компонентов вкладки Additional 3. Использование диалоговых компонентов	Устный опрос. Контрольная работа
3.	Разработка графического интерфейса пользователя Рекомендации: обратить внимание на меню команд Delphi и компоненты обработки событий.	Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам: 1. Многооконные приложения. Обработка событий клавиатуры и "мыши" 2. Технология Drag&Drop	Устный опрос. Контрольная работа
4.	Графика и мультимедиа Рекомендации: обратить внимание на меню команд Delphi и инструменты построения графики, мультимедиа и анимации.	Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам: 1. Построение графических изображений. 2. Мультимедиа и анимация	Устный опрос. Контрольная работа
5	Развитые элементы приложений Рекомендации: обратить внимание на компоненты среды Delphi.	Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам: Разработка Help-системы Разработка и добавление компонентов	Устный опрос Контрольная работа

## 10. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа студентов предполагает тщательное освоение учебной и научной литературы по изучаемой дисциплине.

При изучении основной рекомендуемой литературы студентам необходимо обратить внимание на выделение основных понятий, их определения, научно-технические основы, узловые положения, представленные в изучаемом тексте.

При самостоятельной работе студентов с дополнительной литературой необходимо выделить аспект изучаемой темы (что в данном материале

относится непосредственно к изучаемой теме и основным вопросам).

Дополнительную литературу целесообразно прорабатывать после основной, которая формирует базис для последующего более глубокого изучения темы. Дополнительную литературу следует изучать комплексно, рассматривая разные стороны изучаемого вопроса. Обязательным элементом самостоятельной работы студентов с литературой является ведение необходимых записей: конспекта, выписки, тезисов, планов.

Для самостоятельной работы по дисциплине используется следующее учебно-методическое обеспечение:

а) основная литература

1. Объектно-ориентированное программирование с примерами на C#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с. - Режим доступа: <http://znanium.com/catalog/product/529350>

2. Основы алгоритмизации и программирования на Python : учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 343 с. - Режим доступа: <http://znanium.com/catalog/product/772265>

б) дополнительная литература:

1. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .Net: Учебное пособие /В.Н .Шакин, А.В. Загвоздкина, Г.К. Сосновиков - М.: Форум ,ИНФРА-М, 2015. - 400 с. - Режим доступа: <http://znanium.com/catalog/product/501448>

## **11. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)**

а) основная литература

1. Объектно-ориентированное программирование с примерами на C#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с. - Режим доступа: <http://znanium.com/catalog/product/529350>

2. Основы алгоритмизации и программирования на Python : учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 343 с. - Режим доступа: <http://znanium.com/catalog/product/772265>

б) дополнительная литература:

1. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .Net: Учебное пособие /В.Н .Шакин, А.В. Загвоздкина, Г.К. Сосновиков - М.: Форум ,ИНФРА-М, 2015. - 400 с. - Режим доступа: <http://znanium.com/catalog/product/501448>

## **12. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины (модуля)**

- <https://www.book.ru/> - ЭБС Book.ru
- <http://www.iprbookshop.ru> - ЭБС IPRbooks
- <https://ibooks.ru/> -ЭБС Айбукс.ru/ibooks.ru

- <https://rucont.ru/> - ЭБС «Национальный цифровой ресурс «Руконт»
- <http://znanium.com/> - ЭБС Znanium.com
- <https://dlib.eastview.com/> - База данных East View

**13. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

1. Desktop School ALNG LicSAPk MVL.
  - a. Office ProPlus All Lng Lic/SA Pack MVL Partners in Learning (лицензия на пакет Office Professional Plus)
  - b. Windows 8
2. Система тестирования INDIGO.
3. Adobe Acrobat – свободно-распространяемое ПО
4. Интернет-браузеры Google Chrome, Firefox – свободно-распространяемое ПО
5. Консультант + версия проф.- справочная правовая система

Каждый обучающийся в течение всего обучения обеспечивается индивидуальным неограниченным доступом электронно-библиотечной системе и электронной информационно-образовательной среде.

**14. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)**

Образовательный процесс обеспечивается специальными помещениями, которые представляют собой аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы студентов и помещения для хранения и профилактического обслуживания учебного оборудования.

Специальные помещения соответствуют действующим противопожарным правилам и нормам, укомплектованы специализированной мебелью.

Аудитории лекционного типа, оснащенные проекционным оборудованием и техническими средствами обучения, обеспечивающими представление учебной информации большой аудитории, демонстрационным оборудованием.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, обеспечивающей доступ к сети Интернет и электронной информационно-образовательной среде университета.

## **15. Методические указания для обучающихся по освоению дисциплины (модуля)**

Дисциплина «Объектно-ориентированный анализ и программирование» состоит из 5 тем и изучается на лекциях, лабораторных занятиях и при самостоятельной работе обучающихся. Обучающийся для полного освоения материала должен не пропускать занятия и активно участвовать в учебном процессе. Кроме того, обучающиеся должны ознакомиться с программой дисциплины и списком основной и дополнительной рекомендуемой литературы.

Основной теоретический материал дается на лекционных занятиях. Лекции включают все темы и основные вопросы объектно-ориентированного анализа и программирования. Для максимальной эффективности изучения необходимо постоянно вести конспект лекций, знать рекомендуемую преподавателем основную и дополнительную учебную литературу, позволяющую дополнить знания и лучше подготовиться к практическим занятиям.

Для закрепления теоретического материала, формирования профессиональных компетенций и практических навыков со студентами бакалавриата проводятся лабораторные занятия.

На изучение каждой темы выделено в соответствии с рабочей программой дисциплины количество часов лабораторных занятий, которые проводятся в соответствии с вопросами, рекомендованными к изучению по определенным темам. Обучающиеся должны регулярно готовиться к лабораторным занятиям. При подготовке к занятиям следует руководствоваться конспектом лекций и рекомендованной литературой.

Для эффективного освоения материала дисциплины учебным планом предусмотрена самостоятельная работа, которая должна выполняться в обязательном порядке. Выполнение самостоятельной работы по темам дисциплины, позволяет регулярно проводить самооценку качества усвоения материалов дисциплины и выявлять аспекты, требующие более детального изучения. Задания для самостоятельной работы предложены по каждой из изучаемых тем и должны готовиться индивидуально и к указанному сроку. По необходимости студент бакалавриата может обращаться за консультацией к преподавателю. Выполнение заданий контролируется и оценивается преподавателем.

В случае посещения обучающегося лекций и лабораторных занятий, изучения рекомендованной основной и дополнительной учебной литературы, а также своевременного и самостоятельного выполнения заданий, подготовка к экзамену по дисциплине сводится к дальнейшей систематизации полученных знаний, умений и навыков.

## 16. Методические рекомендации по организации изучения дисциплины для преподавателей, образовательные технологии

Оценочные средства для контроля успеваемости и результатов освоения дисциплины (модуля):

а) для текущей успеваемости: опрос, тесты, лабораторные работы, контрольная работа;

б) для самоконтроля обучающихся: тесты;

в) для промежуточной аттестации: вопросы для экзамена.

При реализации различных видов учебной работы по дисциплине «Объектно-ориентированный анализ и программирование», оценка возможных последствий и контроль над исполнением» используются следующие образовательные технологии:

1) лекции с использованием интерактивных технологий (презентаций).

2) проведение практико-ориентированных занятий (лабораторные работы)

№	Занятие в интерактивной форме	Количество часов по очной форме		Количество часов по заочной форме	
		Лекция	Лаборат.	Лекция	Лаборат.
2	Визуальное программирование Виды: Лекция с демонстрацией видеоматериалов (слайды) Лабораторные занятия с применением следующих технологий: - защита лабораторной работы	2	-	-	-
3	Разработка графического интерфейса пользователя Виды: Лекция с демонстрацией видеоматериалов (слайды) Лабораторные занятия с применением следующих технологий: - защита лабораторной работы	2	-	-	3
4.	Графика и мультимедиа Лабораторные занятия с применением следующих технологий: - защита лабораторной работы	-	2	-	-
5.	Развитые элементы приложений Лабораторные занятия с применением следующих технологий: - защита лабораторной работы	-	2	-	-
<b>Итого</b>		<b>4</b>	<b>4</b>	<b>-</b>	<b>3</b>

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ  
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО  
ДИСЦИПЛИНЕ (МОДУЛЮ)**

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И  
ПРОГРАММИРОВАНИЕ**

Направление подготовки: 38.03.05 Бизнес–информатика  
Направленность: «Электронный бизнес»

## 1. Паспорт фонда оценочных средств

### 1.1. Компетенции, формируемые в процессе изучения дисциплины

Индекс	Формулировка компетенции
ОПК-3	способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях
ПК-18	способность использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования

### 1.2. Сведения об иных дисциплинах (преподаваемых, в том числе, на других кафедрах) участвующих в формировании данных компетенций

#### *1.2.1. Компетенция ОПК-3 формируется в процессе изучения дисциплин (прохождения практик):*

Теоретические основы информатики  
Программирование  
Вычислительные системы, сети, телекоммуникации  
Информационные системы управления производственной компанией  
Управление ИТ-сервисами и контентом  
Компьютерная графика и дизайн  
Учебная практика. Практика по получению первичных профессиональных умений и навыков проектной деятельности, в том числе первичных умений и навыков научно-исследовательской деятельности

#### *1.2.2. Компетенция ПК-18 формируется в процессе изучения дисциплин (прохождения практик):*

Линейная алгебра  
Математический анализ  
Теория вероятностей и математическая статистика  
Программирование  
Дискретная математика  
Информационные технологии анализа и оценки электронного бизнеса  
Производственная практика. Преддипломная практика  
Учебная практика. Практика по получению первичных профессиональных умений и навыков проектной деятельности, в том числе первичных умений и навыков научно-исследовательской деятельности

### 1.3. Этапы формирования и программа оценивания контролируемой компетенции

№	Код контролируемой компетенции	Контролируемые темы дисциплины	Наименование оценочного средства
1 2	ОПК-3 ПК-18	Объектно-ориентированный подход к проектированию и разработке программ Визуальное программирование Разработка графического интерфейса пользователя Графика и мультимедиа Развитые элементы приложений	Тесты. Опрос. Выполнение лабораторных работ Контрольная работа

#### Процедура оценивания

1. Процедура оценивания результатов освоения программы учебной дисциплины включает в себя оценку уровня сформированности компетенций студента при осуществлении текущего контроля и проведении промежуточной аттестации.

2. Уровень сформированности компетенции (ОПК-3, ПК-18) определяется по качеству выполненной студентом работы и отражается в следующих формулировках: высокий, хороший, достаточный, недостаточный.

3. При выполнении студентами заданий текущего контроля и промежуточной аттестации оценивается уровень обученности «знать», «уметь», «владеть» в соответствии с запланированными результатами обучения и содержанием рабочей программы дисциплины:

- профессиональные знания студента могут проверяться при ответе на теоретические вопросы, выполнении тестовых заданий, практических работ,

- степень владения профессиональными умениями – при решении ситуационных задач, выполнении практических работ и других заданий.

4. Результаты выполнения заданий фиксируются в баллах. Общее количество баллов (макс. - 15 б.) складывается из:

- 5 баллов (33,3% от общей оценки) за выполнение практических заданий на выявление уровня обученности «уметь»,

- 5 баллов (33,3% от общей оценки) за выполнение практических заданий на выявление уровня обученности «владеть»,

- 3 балла (20% оценки) за ответы на теоретические вопросы,

- 2 балла (13,3% оценки) за ответы на дополнительные вопросы.

5. По итогам текущего контроля и промежуточной аттестации в соответствии с показателями и критериями оценивания компетенций определяется уровень сформированности компетенций студента и выставляется оценка по шкале оценивания.

### 1.4. Показатели и критерии оценивания компетенций, шкала

Компетенции	Показатели оценивания	Критерии оценивания компетенций				Итого:
		Высокий (верно и в полном объеме) 5 б.	Средний (с незначительными замечаниями) 4 б.	Низкий (на базовом уровне, с ошибками) 3 б.	Недостаточный (содержит большое количество ошибок/ответ не дан) – 2 б.	
<i>Теоретические показатели</i>						
ОПК-3 ПК-18	Знает основные методы анализа и технологии программирования. <hr/> <i>Опрос, тесты</i>	Верно, и в полном объеме знает основные методы анализа и технологии программирования.	С незначительными замечаниями знает основные методы анализа и технологии программирования.	На базовом уровне, с ошибками знает основные методы анализа и технологии программирования.	Не знает основные методы анализа и технологии программирования.	10
	Знает основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций <hr/> <i>Опрос, тесты</i>	Верно, и в полном объеме знает основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций	С незначительными замечаниями знает основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций	На базовом уровне, с ошибками знает основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций	Не знает основные методы и подходы к поиску, сбору, обработке, анализу и систематизации информации, использованию компьютера и глобальных компьютерных сетей для подготовки обзоров, отчетов и научных публикаций	
<i>Практические показатели</i>						
ОПК-3 ПК-18	Умеет разрабатывать алгоритмы решения,	Верно, и в полном объеме может разрабатывать алгоритмы	С незначительными замечаниями может разрабатывать алгоритмы решения,	На базовом уровне, с ошибками может разрабатывать алгоритмы решения, программировать	Не может разрабатывать алгоритмы решения,	10

Компетенции	Показатели оценивания	Критерии оценивания компетенций				Итого:
		Высокий (верно и в полном объеме) 5 б.	Средний (с незначительными замечаниями) 4 б.	Низкий (на базовом уровне, с ошибками) 3 б.	Недостаточный (содержит большое количество ошибок/ответ не дан) – 2 б.	
	<p>программировать задачи обработки данных в предметной области; выполнять тестирование и отладку программ.</p> <hr/> <p><i>Лабораторная работа</i></p>	решения, программировать задачи обработки данных в предметной области; выполнять тестирование и отладку программ.	программировать задачи обработки данных в предметной области; выполнять тестирование и отладку программ.	задачи обработки данных в предметной области; выполнять тестирование и отладку программ.	программировать задачи обработки данных в предметной области; выполнять тестирование и отладку программ.	
	<p>Умеет применять полученные в процессе обучения знания при решении задач профессиональной деятельности;</p> <hr/> <p><i>Лабораторная работа</i></p>	Верно, и в полном объеме может применять полученные в процессе обучения знания при решении задач профессиональной деятельности;	С незначительными замечаниями может применять полученные в процессе обучения знания при решении задач профессиональной деятельности;	На базовом уровне, с ошибками может применять полученные в процессе обучения знания при решении задач профессиональной деятельности;	Не может применять полученные в процессе обучения знания при решении задач профессиональной деятельности;	
<i>Владеет</i>						
ОПК-3 ПК-18	<p>Владеет методами и инструментальными средствами разработки программ.</p> <hr/> <p><i>Контрольная работа</i></p>	Верно, и в полном объеме владеет методами и инструментальными средствами разработки программ	С незначительными замечаниями владеет методами и инструментальными средствами разработки программ	На базовом уровне, с ошибками владеет методами и инструментальными средствами разработки программ	Не владеет методами и инструментальными средствами разработки программ	10
	Владеет	Верно, и в полном	С незначительными	На базовом уровне, с	Не владеет	

Компетенции	Показатели оценивания	Критерии оценивания компетенций				Итого:
		Высокий (верно и в полном объеме) 5 б.	Средний (с незначительными замечаниями) 4 б.	Низкий (на базовом уровне, с ошибками) 3 б.	Недостаточный (содержит большое количество ошибок/ответ не дан) – 2 б.	
	информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;  <i>Контрольная работа</i>	объемом владеет информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;	замечаниями владеет информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;	ошибками владеет информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;	информационными компьютерными технологиями, необходимыми при выполнении научно-исследовательской работы;	
						Всего 30

### Шкала оценивания:

Оценка	Баллы	Уровень сформированности компетенции
отлично	26-30	высокий
хорошо	21-25	хороший
удовлетворительно	15-20	достаточный
неудовлетворительно	14 и менее	недостаточный

## **2. Типовые контрольные задания для оценки результатов обучения по дисциплине и иные материалы для подготовки к промежуточной аттестации**

### **2.1. Материалы для подготовки к промежуточной аттестации**

#### **Вопросы к экзамену:**

1. ИСР Visual Basic Application (VBA), основные элементы.
2. Проект VCL Forms Application. Файлы, входящие в проект.
3. Консольное приложение, его отличие от windows-приложения.
4. Понятие класса, объекта, метода, конструктора, деструктора.
5. Понятие инкапсуляции, свойства.
6. Возможности переопределения операций.
7. Понятие наследования, абстрактные классы.
8. Описание и использование динамических массивов.
9. Строки. Функции для работы со строками.
10. Списки строк (TStrings), их методы и свойства.
11. Форма, ее свойства. Понятие события.
12. Свойства и события компонентов TLabel, TStaticText.
13. Свойства и события компонентов TEdit, TMaskEdit, TUpDown.
14. Свойства и события компонентов TListBox, TComboBox, TCheckListBox.
15. Свойства и события компонентов TMainMenu, TPopupMenu.
16. Свойства и события компонентов TMemo, TRichEdit.
17. Свойства и события компонентов TButton, TBitBtn, TSpeedButton.
18. Свойства и события компонентов TCheckBox, TRadioButton, TRadioGroup.
19. Свойства и события компонентов TScrollBar, TScrollBox.
20. Свойства и события компонентов TGroupBox, TPanel.
21. Свойства и события компонентов TStringGrid, TDrawGrid.
22. Свойства и события компонентов TImage, TImageList, TShape.
23. Свойства и события компонентов TOpenDialog, TSaveDialog, TOpenPictureDialog, TSavePictureDialog.
24. Свойства и события компонентов TColorDialog, TFontDialog.
25. Свойства и события компонентов TReplaceDialog, TFindDialog.
26. Свойства и события компонентов TPrintDialog, TPrinterSetupDialog, TPageSetupDialog.
27. Модальная форма, ее вывод, особенности управления.
28. События, связанные с клавиатурой.
29. События, связанные с "мышью".
30. События и методы, реализующие технологию Drag&Drop.
31. Особенности использования фреймов.
32. Использование компонента TTimer, THotKey.
33. Свойства и события компонентов TTabControl, TPageControl.

34. Свойства и события компонентов TValueListEditor, TColorBox.
35. Свойства и события компонентов TDateTimePicker, TMonthCalendar.
36. Свойства и события компонентов THeaderControl, TStatusBar.
37. Свойства и события компонентов TToolBar, TCoolBar, TPageScroller.
38. Свойства и события компонентов TTrackBar, TProgressBar.
39. Свойства и события компонентов TreeView, ListView.
40. Свойства и события компонента ComboBoxEx.
41. События компонента TApplicationEvents.
42. Использование компонента TChart.
43. Понятие канвы. Рисование по пикселям. Событие OnPaint.
44. Перо, его методы и свойства.
45. Кисть, ее методы и свойства.
46. Классы для хранения графических объектов.
47. Компоненты TMediaPlayer, TAnimate.
48. Средства создания hlp- и chm-файлов.
49. Этапы разработки собственных и подключения сторонних компонентов.
50. Средства создания библиотек DLL и их динамическое подключение.
51. Средства порождения дочерних процессов.
52. Компоненты для работы с серверами MS Office.
53. Подключение к серверам MS Office.
54. Свойства, методы сервера MS Word.
55. Свойства, методы сервера MS Excel.

### **Типовые контрольные задания:**

1. С использованием компонентов Label, Edit, Мемо и Button составить программу, вычисляющую корни квадратного уравнения.
2. С использованием компонентов Label, Edit, Мемо и Button составить программу, вычисляющую максимальный из трех введенных чисел
3. С использованием компонентов Label, Edit, Мемо и Button составить программу, имитирующую работу калькулятора.

### **Образцы тестовых заданий для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины, а также для контроля самостоятельной работы:**

1. Логическое выражение  
 $(N \text{ Mod } 10 \text{ Mod } 2) \text{ Or } (N \text{ Div } 10 \text{ Mod } 10 \text{ Mod } 2 = 0) \text{ Or } (N \text{ Div } 100 \text{ Mod } 2 = 0)$   
 должно принимать значение TRUE тогда и только тогда, когда истинно высказывание

- а) в трёхзначном натуральном числе все цифры чётные;
- б) в трёхзначном натуральном числе одна чётная цифра;
- в) в трёхзначном натуральном числе две чётных цифры;
- г) в трёхзначном натуральном числе хотя бы одна чётная цифра;
- д) в трёхзначном натуральном числе нет чётных цифр.

2. Ошибку "Structure too large" (структура превышает максимально допустимый объём статической памяти) вызовет описание

- а) Type Vector = Array[Byte] Of Integer; Var C : Array[1..10] Of Vector;
- б) Var T : File Of String;
- в) Type A = Record S : String; A, B, C : Array[10..20] Of Real End;
- г) Var M : Array[1..5, 1..8] Of A;
- д) Var K : Array [Byte, Byte] Of String[6];
- е) Var S : Array[-10000..10000] Of String[2].

3. К процедурам для работы с динамическими переменными не относится

- а) Mark;
- б) New;
- в) Release;
- г) Seek;
- д) Dispose.

4. Имеется описание

Type A = Array[0..100] Of Real; B = ^A; Var M : Array[1..5] Of B;

Для хранения массива M необходим объём памяти (байт)

- а) 606;
- б) 4;
- в) 20;
- г) 12120;
- д) 6.

5. Фрагмент программы

K := 0;

While Not Eof(F) Do

  Begin ReadLn(F, S); I := 1;

  While I <= Length(S) Do

    Begin If S[I] In ['A'..'Я', 'a'..'п', 'р'..'я']

      Then Begin K := K + 1;

        Delete(S, I, 1); I:= I - 1

      End;

    I := I + 1

  End

End;

выполняет следующее действие:

- а) удаляет из текстового файла F все русские буквы;
- б) определяет в текстовом файле количество символов, являющихся русскими буквами;
- в) определяет в текстовом файле количество символов, не являющихся русскими буквами;
- г) определяет в текстовом файле количество символов;
- д) удаляет из текстового файла F все символы, не являющиеся русскими буквами.

6. В фрагменте программы (здесь Var F : File Of Integer; I, K, Vsp : Integer;)

Reset(F); K := FileSize(F) – 1;

For I := 0 To K Do

Begin Seek(F, I); Read(F, Vsp); Seek(F, FileSize(F)); Write(F, Vsp) End;

выполняется

- а) сортировка файла;
- б) изменение порядка следования элементов на обратный;
- в) дописывание в конец исходного файла полную его копию с сохранением порядка следования элементов;
- г) дописывание в конец исходного файла полную его копию с изменением порядка следования элементов на противоположный;
- д) не выполняется никаких действий по изменению файла.

7. Имеется описание

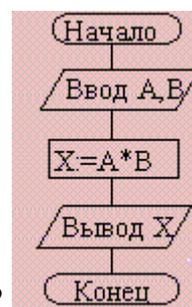
Type Dn = (pn, vt, sr, cht, ptn, sb, vs); Mn = Set Of Dn; Var V : Mn;

и фрагмент программы

V := [pn..ptn] \* [sr, ptn..vs] – [sb];

После исполнения этого фрагмента переменная V имеет значение

- а) [pn..vs];
- б) [sr, ptn];
- в) [sb];
- г) [];
- д) [pn..ptn].

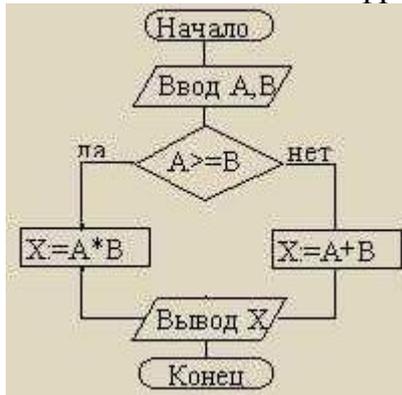


8. Алгоритм какого типа изображен на блок-схеме?

- а) циклический;
- б) разветвляющийся;
- в) вспомогательный;
- г) линейный;

д) комбинация развилки и цикла.

9. После исполнения фрагмента программы, изображенного на блок-



схеме,

при  $A = 5$ ,  $B = 4$  значение  $X$  будет равно

- а) 20;
- б) 9;
- в) 5;
- г) 4;
- д) 1.

10. В приведенном фрагменте программы ( $N$  типа `LongInt`,  $N > 0$ )

$P := 1$ ;

While  $P \leq N$  Do

Begin

Left :=  $N \text{ Div } (P * 10) * (P * 10)$ ;

Right :=  $N \text{ Mod } P$ ;

$K := ((N \text{ Mod } (P * 10) \text{ Div } P + 1) \text{ Mod } 10) * P$ ;

$N := \text{Left} + K + \text{Right}$ ;  $P := P * 10$

End;

натуральное число  $N$  изменяется по следующему правилу

- а) не изменяется;
- б) в каждый разряд прибавляется 1;
- в) из каждого разряда вычитается 1;
- г) в каждый разряд прибавляется 1, если значение в разряде — не девять, иначе заменяется на нуль;
- д) каждая девятка в десятичной записи числа заменяется на нуль.

11. Цикл с предусловием выполняется так:

- а) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла;
- б) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла;
- в) проверяется условие продолжения выполнения цикла, выполняется тело цикла;
- г) тело цикла выполняется  $N$  раз ( $N$  — натуральное);

д) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

12. В текстовом файле каждая строка заканчивается

- а) числами 10 и 13;
- б) символами с кодами 10 и 13;
- в) символом с кодом 13;
- г) числом 0;
- д) символом с кодом 10.

13. Процедуры ReadLn и WriteLn можно использовать при работе с

- а) типизированными файлами;
- б) нетипизированными файлами;
- в) типизированными и нетипизированными файлами;
- г) текстовыми файлами;
- д) любыми файлами.

14. Значение выражения  $\text{Ord}(x > y) + \text{Ord}(\text{Ord}(z = 'F'))$  при  $x = 7, y = 0, z = 'F'$  равно

- а) TRUE;
- б) FALSE;
- в) 0;
- г) 1;
- д) 2.

15. Идентификатор в Turbo Pascal не может начинаться с

- а) латинской буквы;
- б) заглавной латинской буквы;
- в) цифры;
- г) знака подчёркивания;
- д) латинской буквы, а затем знака подчёркивания.

### **Литература для подготовки к экзамену:**

а) основная литература

1. Объектно-ориентированное программирование с примерами на C#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с. - Режим доступа: <http://znanium.com/catalog/product/529350>

2. Основы алгоритмизации и программирования на Python : учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2017. — 343 с. - Режим доступа: <http://znanium.com/catalog/product/772265>

б) дополнительная литература:

1. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .Net: Учебное пособие /В.Н .Шакин, А.В. Загвоздкина, Г.К. Сосновиков - М.: Форум ,ИНФРА-М, 2015. - 400 с. - Режим доступа: <http://znanium.com/catalog/product/501448>

## **Промежуточная аттестация**

### **2.2. Комплект экзаменационных билетов для проведения промежуточной аттестации**

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность: «Электронный бизнес»

Дисциплина: «Объектно-ориентированный анализ и программирование»

#### **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1**

1. Понятие класса, объекта, метода, конструктора, деструктора.
2. Свойства и события компонентов TEdit, TMaskEdit, TUpDown.

#### **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2**

1. Описание и использование динамических массивов.
2. Свойства и события компонентов TMainMenu, TPopupMenu.

#### **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3**

1. Понятие наследования, абстрактные классы.
2. Свойства и события компонентов TDateTimePicker, TMonthCalendar.

#### **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4**

1. Средства создания библиотек DLL и их динамическое подключение.
2. Свойства и события компонентов TPrintDialog, TPrinterSetupDialog, TPageSetupDialog.

#### **ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 5**

1. Понятие инкапсуляции, свойства.
2. Свойства и события компонентов TToolBar, TCoolBar, TPageScroller.

**Промежуточная аттестация**  
**Комплект тестовых заданий для проведения экзамена по дисциплине**

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

Направление подготовки: 38.03.05 Бизнес-информатика  
Направленность: «Электронный бизнес»  
Дисциплина: «Объектно-ориентированный анализ и программирование»

Тестовые задания для проверки уровня обученности **ЗНАТЬ:**

1. Ввод данных — это
  - а) процесс передачи данных из оперативной памяти на внешний носитель;
  - б) процесс ввода с клавиатуры каких-либо значений;
  - в) передача данных от внешнего носителя в оперативную память для обработки;
  - г) присваивание конкретных значений переменным, которые используются в программе;
  - д) запись файла на диск.
  
2. Свойством алгоритма является
  - а) результативность;
  - б) цикличность;
  - в) возможность изменения последовательности выполнения команд;
  - г) возможность выполнения алгоритма в обратном порядке;
  - д) простота при записи на языках программирования.
  
3. Из перечисленных ниже в программе обязательны
  - а) раздел Var;
  - б) раздел Const;
  - в) раздел Type;
  - г) раздел Label;
  - д) раздел Begin ... End.
  
4. Среди перечисленных соответствий, которые необходимо соблюдать между формальными и фактическими параметрами лишним является
  - а) соответствие по типу параметров;
  - б) соответствие по количеству параметров;

- в) соответствие по типу используемых вспомогательных переменных;
- г) соответствие по порядку перечисления;

5. В приведенном фрагменте программы (First — ссылка на первый элемент списка; список объявлен следующим образом: Type SS = ^List; List = Record A : LongInt; Next : SS End;)

```
P := First; S := 0; While Not (P = Nil) Do Begin S := S + 1; P := P^.Next
End;
```

определяется

- а) первый элемент списка;
- б) сумма элементов списка;
- в) сумма первого и последнего элементов списка;
- г) количество элементов списка;
- д) количество звеньев списка, где указатель на следующее звено не

Nil.

Тестовые задания для проверки уровня обученности **УМЕТЬ, ВЛАДЕТЬ:**

1. Значения переменных  $a$  и  $b$  после выполнения следующих действий  
 $a := 15 \text{ Div } (16 \text{ Mod } 7)$ ;  $b := 34 \text{ Mod } a * 5 - 29 \text{ Mod } 5 * 2$ ;

будут равны

- а)  $a = 1, b = 160$ ;
- б)  $a = 1, b = 4$ ;
- в)  $a = 7, b = 25$ ;
- г)  $a = 7, b = 22$ ;
- д)  $a = 7, b = 28$ .

2. Во фрагмент алгоритма

```
For K := 10 To 99 Do
```

```
Begin P1 := K Div 10;
```

```
P2 := K Mod 10;
```

```
S := P1 + P2;
```

```
If ____ Then WriteLn(K)
```

```
End;
```

печатающий все двузначные числа, в записи которых есть цифра  $N$  или сумма цифр которых равна самим числам, нужно вписать логическое выражение

- а)  $(P1 = N) \text{ Or } (P2 = N) \text{ And } (S = K)$ ;
- б)  $(P1 = N) \text{ Or } (P2 = N) \text{ Or } (S = K)$ ;
- в)  $(P1 = N) \text{ And } (P2 = N) \text{ Or } (S = K)$ ;
- г)  $((P1 = N) \text{ Or } (P2 = N)) \text{ And } (S = K)$ ;
- д)  $(P1 = N) \text{ And } (P2 = N) \text{ And } (S = N)$ .

3. Какая из приведенных серий операторов определяет и печатает индекс последнего отрицательного элемента в линейном массиве из  $n$  элементов?

1)  $i := n$ ; While ( $i \geq 1$ ) And ( $m[i] > 0$ ) Do Dec ( $i$ ); If  $i < 1$  Then WriteLn (' $i = 0$ ') Else WriteLn (' $i =$ ',  $i$ );

2)  $k := 0$ ; For  $i := 1$  To  $n$  Do If  $m[i] < 0$  Then  $k := i$ ; WriteLn (' $i =$ ',  $k$ );

3)  $i := n$ ; Repeat  $i := i - 1$  Until ( $m[i] < 0$ ); WriteLn (' $i =$ ',  $i$ );

а) 1,2;

б) 2, 3;

в) 1, 2,3;

г) 1;

д) ни один из ответов 1–4 не верен.

4. Задан двумерный массив  $X[1..n, 1..m]$ . Процедура

Procedure Sub (Var X: Myarray);

Var i, j: Integer;

Begin For  $i := 1$  To  $n$  Do

For  $j := 1$  To  $m \text{ Div } 2$  Do  $X[i, 2 * j] := X[i, 2 * j] + X[i, 1]$ ;

End;

а) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки;

б) добавляет к матрице еще  $M$  столбцов с элементами, равными соответствующим элементам первого столбца;

в) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки;

г) к элементам четных строк прибавляет элементы первой строки соответствующего столбца;

д) меняет порядок столбцов таблицы.

5. С помощью какой из приведенных серий команд переменной  $B$  присваивается значение выражения?

$$\left( \frac{x+y}{x-y} - \frac{x-y}{x+y} \right) \cdot \left( \frac{x-y}{x+y} + \frac{x+y}{x-y} \right)$$

а)  $A := (x + y) / (x - y)$ ;  $B := (A - 1 / A) * (1 / A + A)$ ;

б)  $A := (x + y) / (x - y)$ ;  $B := \text{Sqr}(A) - \text{Sqr}(1 / A)$ ;

в)  $A := (x - y) / (x + y)$ ;  $B := \text{Sqr}(1 / A) - \text{Sqr}(A)$ ;

### 2.3. Критерии оценки для проведения экзамена по дисциплине

После завершения тестирования на экзамене на мониторе компьютера высвечивается результат – процент правильных ответов. Результат переводится в баллы и суммируется с текущими семестровыми баллами.

Максимальная сумма (100 баллов), набираемая студентом по дисциплине, предусматривающей в качестве формы промежуточной аттестации экзамен, включают две составляющие.

Первая составляющая – оценка регулярности и своевременности качества выполнения студентом учебной работы по изучению дисциплины в течение семестра (сумма не более 60 баллов).

Вторая составляющая – оценка знаний студента на экзамене (не более 40 баллов).

Перевод полученных итоговых баллов в оценки осуществляется по следующей шкале:

- с 86 до 100 баллов – «отлично»;
- с 71 до 85 баллов – «хорошо»;
- с 50 до 70 баллов – «удовлетворительно»

Если студент при тестировании отвечает правильно менее, чем на 50 %, то автоматически выставляется оценка «неудовлетворительно» (без суммирования текущих рейтинговых баллов), а студенту назначается переэкзаменовка в дополнительную сессию.

#### **2.4. Методические материалы, определяющие процедуру оценивания по дисциплине**

Общая процедура оценивания определена Положением о фондах оценочных средств.

1. Процедура оценивания результатов освоения программы дисциплины включает в себя оценку уровня сформированности общекультурных и профессиональных компетенций студента, уровней обученности: «знать», «уметь», «владеть».

2. При сдаче экзамена:

– профессиональные знания студента могут проверяться при ответе на теоретические вопросы, при выполнении тестовых заданий, практических работ;

– степень владения профессиональными умениями, уровень сформированности компетенций (элементов компетенций) – при решении ситуационных задач, выполнении практических работ и других заданий.

3. Результаты промежуточной аттестации фиксируются в баллах. Общее количество баллов складывается из следующего:

- до 60% от общей оценки за выполнение практических заданий,
- до 30% оценки за ответы на теоретические вопросы,
- до 10% оценки за ответы на дополнительные вопросы.

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО  
КОНТРОЛЯ И ТЕКУЩЕЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ  
(МОДУЛЮ)**

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЙ АНАЛИЗ И  
ПРОГРАММИРОВАНИЕ**

Направление подготовки: 38.03.05 «Бизнес-информатика»  
Направленность: «Электронный бизнес»

## 1. Материалы для текущего контроля

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

### Вопросы для проведения опроса

по дисциплине «Объектно-ориентированный анализ и программирование»

1. Интегрированная среда разработки (ИСП) – структура, возможности. Проект, файлы, входящие в состав проекта.
2. Программирование под Windows.
3. Библиотека функций Windows API.
4. Основные принципы объектно-ориентированного программирования. Особенности языка программирования Turbo Delphi (Visual Basic).
5. Классы и объекты, поля, свойства, методы, события.
6. Конструкторы и деструкторы.
7. Форма: свойства и методы формы.
8. События, организация реакции на них.
9. Визуальные компоненты, обзор, использование библиотек. Компоненты ввода и отображения текстовой информации.
10. Компоненты – элементы управления.
11. Компоненты - меню.
12. Компоненты внешнего оформления.
13. Организация диалогов.
14. Компоненты – диалоги.
15. Требования к интерфейсу, многооконные приложения.
16. События клавиатуры и "мыши".
17. Технология Drag&Drop. "
18. Продвинутое" компоненты для организации интерфейса пользователя.
19. Компоненты отображения графической информации.
20. Канва, перо, кисть, их свойства и методы.
21. Мультимедиа и анимация.
22. Помощь пользователю: Help-система, подсказки, строка состояния. Разработка и добавление компонентов..
23. Сообщения Windows и их обработка.
24. COM-технология программирования.
25. Свойства, методы серверов MS Word и MS Excel.

**Критерии оценки:**

- оценка «отлично» выставляется студенту, если он верно и в полном объеме ответил на все поставленные вопросы;
- оценка «хорошо» выставляется студенту, если он ответил на все поставленные вопросы с незначительными замечаниями;
- оценка «удовлетворительно» выставляется студенту, если он ответил на поставленные вопросы на базовом уровне, с ошибками, которые при дополнительных вопросах исправил;
- оценка «неудовлетворительно» выставляется студенту, если он ответил на поставленные вопросы с большим количеством ошибок или ответил неправильно.

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

### Тестовые вопросы

по дисциплине «Объектно-ориентированный анализ и программирование»

1. Тело цикла в программе

```
a:= 1; b:= 1; While a + b < 8 Do begin a := a + 1; b:= b + 2 end;
```

выполнится

- а) 1 раз;
- б) 2 раза;
- в) 3 раза;
- г) ни разу;
- д) бесконечное число раз.

2. Элементы массива  $p[1..5]$  равны соответственно 1, -1, 5, 2, 4.

Значение выражения

$$p[1] * p[3] - p[2 * p[2] + p[p[5] - p[2]]]$$

равно

- а) 8;
- б) -8;
- в) 12;
- г) -12;
- д) 6.

3. Задана строка  $St$ . Фрагмент алгоритма

```
S := 0;
```

```
For I := 1 To Length (St) Do
```

```
Begin
```

```
Val(St[I], d, k);
```

```
If K = 0 Then S := S + d
```

```
End;
```

- а) определяет количество цифр в строке;
- б) подсчитывает количество нулей в строке;
- в) определяет сумму номеров позиций в строке, где стоят цифры;
- г) подсчитывает сумму цифр в строке;
- д) определяет сумму номеров позиций в строке, где стоят нули.

4. Какая из приведенных серий операторов определяет и печатает индекс последнего отрицательного элемента в линейном массиве из  $n$  элементов?

1)  $i := n$ ; While ( $i \geq 1$ ) And ( $m[i] > 0$ ) Do Dec ( $i$ ); If  $i < 1$  Then WriteLn (' $i = 0$ ') Else WriteLn (' $i =$ ',  $i$ );

2)  $k := 0$ ; For  $i := 1$  To  $n$  Do If  $m[i] < 0$  Then  $k := i$ ; WriteLn (' $i =$ ',  $k$ );

3)  $i := n$ ; Repeat  $i := i - 1$  Until ( $m[i] < 0$ ); WriteLn (' $i =$ ',  $i$ );

а) 1, 2;

б) 2, 3;

в) 1, 2, 3;

г) 2;

д) ни один из ответов 1–4 не верен.

5. Задан линейный массив  $M[1..n]$ .

Function Control (M: Myarray): Boolean;

Var I : Integer;

Begin I := 1;

While (I <= n) And (M[I] > 0) Do Inc(I);

Control := (I <= n);

End;

Если в данном массиве все элементы положительные, приведенная функция возвращает значение

а)  $n$ ;

б) True;

в) False;

г)  $I \leq n$ ;

д) ни один из ответов 1–4 не верен.

6. Задан двумерный массив  $X[1..n, 1..m]$ . Процедура

Procedure Sub (Var X: Myarray);

Var i, j: Integer;

Begin For i := 1 To n Do

For j := 1 To m Div 2 Do  $X[i, 2 * j] := X[i, 2 * j] + X[i, 1]$ ;

End;

а) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки;

б) добавляет к матрице еще  $M$  столбцов с элементами, равными соответствующим элементам первого столбца;

в) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки;

г) к элементам четных строк прибавляет элементы первой строки соответствующего столбца;

д) меняет порядок столбцов таблицы.

7. Задан двумерный массив  $X[1..n, 1..m]$ . Функция

Function Check (X: Myarray): Boolean;

Var i, j : Integer; t : Boolean;

Begin t := True; i := 1;

```

While t And (i <= n) Do
Begin j := 1; While (j <= m) And (X[i, j] <> 0) Do Inc (j);
t := (j = m + 1); Inc (i)
End;
Check := Not t
End;

```

возвращает значение

- а) True, если все элементы массива ненулевые;
- б) True, если в массиве есть элемент, равный нулю;
- в) False, если в массиве есть элемент, равный нулю;
- г) Not t;
- д) ни один из ответов 1–4 не верен.

8. Среди перечисленных соответствий, которые необходимо соблюдать между формальными и фактическими параметрами

- а) соответствие по типу параметров;
- б) соответствие по количеству параметров;
- в) соответствие по типу используемых вспомогательных переменных;
- г) соответствие по порядку перечисления;

лишним является

- 1) а; 2) б; 3) в; 4) г; 5) ни один из ответов 1–4 не верен.

9. Определите тип выражения (здесь  $A : \text{Array}[1..20] \text{ Of Real}$ ;  $B : \text{Boolean}$ ;  $C : \text{Integer}$ )

$C + \text{Ord}(\text{Round}(A[7]) + \text{Ord}(B)) - \text{Trunc}(A[1])$

- а) Real;
- б) Integer;
- в) Boolean;
- г) Byte;
- д) среди ответов 1–4 нет верного.

10. Список объявлен следующим образом

Type Ukaz = ^Zveno; Zveno = Record X : String; N : Ukaz End;

Var First : Ukaz; {ссылка на начало списка}

В следующем фрагменте программы

P:= First;

While P^.N <> Nil Do

Begin B:= P; M:= P;

While B <> Nil Do Begin If B^.X < M^.X Then M:= B; B := B^.N End;

S := P^.X; P^.X := M^.X; M^.X := S; P := P^.N

End;

выполняется

- а) перемещение компонента к началу списка;
- б) сортировка компонентов списка в порядке возрастания;
- в) сортировка компонентов списка в порядке убывания;

- г) перестановка соседних компонентов списка;
- д) добавление в список нескольких новых компонент.

11. Фрагмент программы  $S := A; A := B; B := S;$  выполняет

- а) обмен значений переменных  $A, B;$
- б) присваивание переменным  $A, B$  значения  $S;$
- в) замена значения переменной  $A$  значением переменной  $B;$
- г) во фрагменте не выполняется никаких действий;
- д) замена значения переменной  $B$  значением переменной  $A.$

12. Имеется следующее описание

Type U = ^Zveno; Zveno = Record X, Y: Boolean; Pred, Next : U End;  
 Var Logic: Boolean; A, B: Pointer; X, Y: U;

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание

- а)  $A := X^.Next^.Next;$
- б)  $X := Y;$
- в)  $Logic := X^.X;$
- г)  $X^.Next := A;$
- д)  $X^ := Y^.Next.$

13. При присваивании изменяется

- а) алгоритм;
- б) имя переменной;
- в) тип переменной;
- г) значение переменной;
- д) значение константы.

14. Задана строка  $St.$  Фрагмент алгоритма

```
S:= 0;
For I := 1 To Length (St) Do
Begin
Val(St[I], d, k);
If K = 0 Then S:= S + d
End;
```

- а) определяет количество цифр в строке;
- б) подсчитывает количество нулей в строке;
- в) определяет сумму номеров позиций в строке, где стоят цифры;
- г) подсчитывает сумму цифр в строке;
- д) определяет сумму номеров позиций в строке, где стоят нули.

15. Задан двумерный массив  $X[1..n, 1..m].$  Функция

```
Function Check (X: Marray): Boolean;
Var i, j: Integer; t : Boolean;
Begin:= True; i := 1;
```

```
While t And (i <= n) Do
Begin j:= 1; While (j <= m) And (X[i, j] <> 0) Do Inc (j);
t:= (j = m + 1); Inc (i)
End;
Check:= Not t
End;
```

возвращает значение

- а) True, если все элементы массива ненулевые;
- б) True, если в массиве есть элемент, равный нулю;
- в) False, если в массиве есть элемент, равный нулю;
- г) Not t;
- д) ни один из ответов 1–4 не верен.

**Критерии оценки:**

Студент аттестован, если правильно ответил более чем на 10 вопросов.

## 2. Материалы для проведения текущей аттестации

### Текущая аттестация 1

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

### Комплект заданий для проведения текущей аттестации №1 (в форме лабораторной работы)

по дисциплине «Объектно-ориентированный анализ и программирование»

#### Лабораторная работа №1 «Визуальное программирование»

Создать новый проект. Потренируйтесь добавлять и удалять визуальных компонентов на форму. Поработайте со свойствами объектов с помощью инспектора объектов. Добавьте обработчики событий. Запустите программу из интегрированной среды. Выполните пошаговую отладку с инспектированием значений переменных. Потренируйтесь в создании, добавлении и удалении модулей и форм. Выполните настройку опций проекта, а также настройку опций среды программирования. Сохраните проект.

1. Открыть новый проект можно, выбрав пункт меню File | New Project. Для работы можно создать форму согласно образцу (рисунок 1.12).

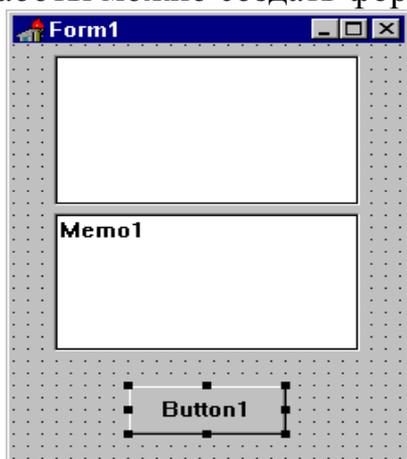


Рисунок 1.12 – Образец формы

2. Поместите на форму объект TMemo, а затем TEdit так, чтобы он наполовину перекрывал TMemo, как показано на рисунке 1.13.

3. Теперь выберите пункт меню Edit | Send to Back, что приведет к перемещению TEdit вглубь формы, за объект TMemo. Это называется

изменением Z-порядка компонент. Поместите кнопку TButton в нижнюю часть формы. Теперь растяните Инспектор Объектов так, чтобы свойства Name и Caption были видны одновременно на экране. Теперь измените имя кнопки на Terminate. Текст, который Вы видите на поверхности кнопки - это значение свойства Caption. Напишите обработчик нажатия на кнопку, который будет закрывать приложение.

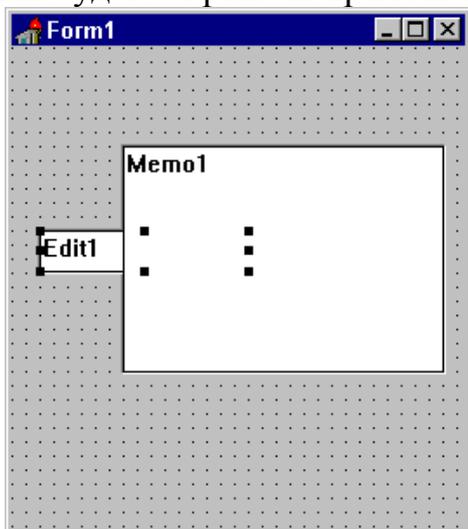


Рисунок 1.13 – Образец формы

Для работы со свойствами нажмите клавишу <Shift> и щелкните на TMemo и затем на TListBox. Теперь оба объекта имеют по краям маленькие квадратики, показывающие, что объекты выбраны. Выбрав два или более объектов одновременно, можно выполнить большое число операций над ними, например, передвигать по форме. Затем попробуйте выбрать пункт меню Edit | Size и установить оба поля Ширину(Width) и Высоту(Height) в Grow to Largest, как показано на рисунке 1.14. Теперь оба объекта стали одинакового размера.

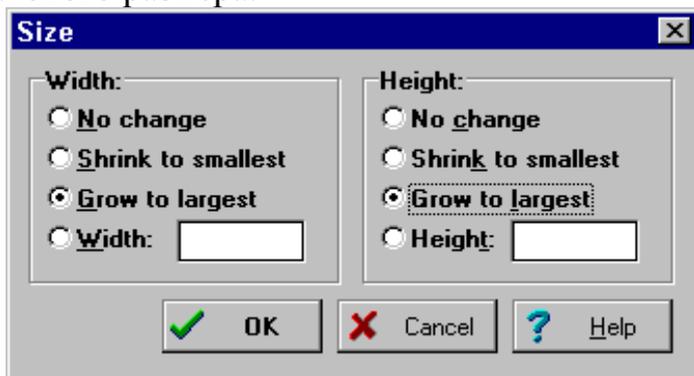


Рисунок 1.14 – Диалог Size

4. Затем выберите пункт меню Edit | Align и поставьте в выравнивании по горизонтали значение Center (рисунок 1.15).

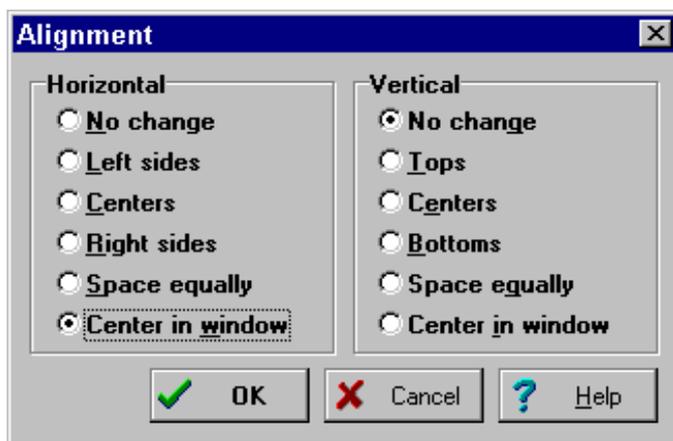


Рисунок 1.15 – Диалог Alignment

5. Поскольку выбрано два компонента, то содержимое Инспектора Объектов изменится - он будет показывать только те поля, которые являются общими для объектов. Это означает то, что изменения в свойствах повлияют не на один, а на все выбранные объекты.

Рассмотрим изменение свойств объектов на примере свойства Color. Есть три способа изменить его значение в Инспекторе Объектов. Первый - просто напечатать имя цвета (clRed) или номер цвета. Второй путь - нажать на маленькую стрелку справа и выбрать цвет из списка. Третий путь - дважды щелкнуть на поле ввода свойства Color. При этом появится диалог выбора цвета.

Свойство Font работает аналогично свойству Color.

Дважды щелкните на свойство Items объекта ListBox. Появится диалог, в котором Вы можете ввести строки для отображения в ListBox.

Напечатайте несколько слов, по одному на каждой строке, и нажмите кнопку ОК. Текст отобразится в ListBox'e.

6. Сохранение программы:

Создать поддиректорию для программы. Лучше всего создать директорию, где будут храниться все Ваши программы и в ней создать поддиректорию для данной конкретной программы.

После создания поддиректории для хранения программы нужно выбрать пункт меню File | Save Project. Сохранить нужно будет два файла. Первый - модуль (unit), над которым Вы работали, второй - главный файл проекта, который "владеет" Вашей программой. Сохраните модуль под именем MAIN.PAS и проект под именем TIPS1.DPR.

7. Для разработки обработчика нажатия на кнопку перейдите на форму и дважды щелкните мышкой на объект TButton. Вы попадете в окно Редактора, в котором будет фрагмент кода:

```
procedure TForm1.TerminateClick(Sender: TObject);  
begin  
end;
```

Данный код был создан автоматически и будет выполняться всякий раз, когда во время работы программы пользователь нажмет кнопку Terminate. Определение класса в начале файла теперь включает ссылку на метод TerminateClick:

```
TForm1 = class(TForm)
Edit1: TEdit;
Memo1: TMemo;
Terminate: TButton;
procedure TerminateClick(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
```

8. Для того, чтобы приложение закрывалось при нажатии на кнопку необходимо написать код:

```
procedure TForm1.TerminateClick(Sender: TObject);
begin
Close;
end;
```

## Лабораторная работа №2 «Визуальное программирование»

Изучение основных визуальных компонентов по страницам. Приобретение навыков использования визуальных компонентов из страниц: Standard, Additional, Win32, Dialogs, Samples.

1. Разработать программу с интерфейсом, приведенным на рисунке 2.1.

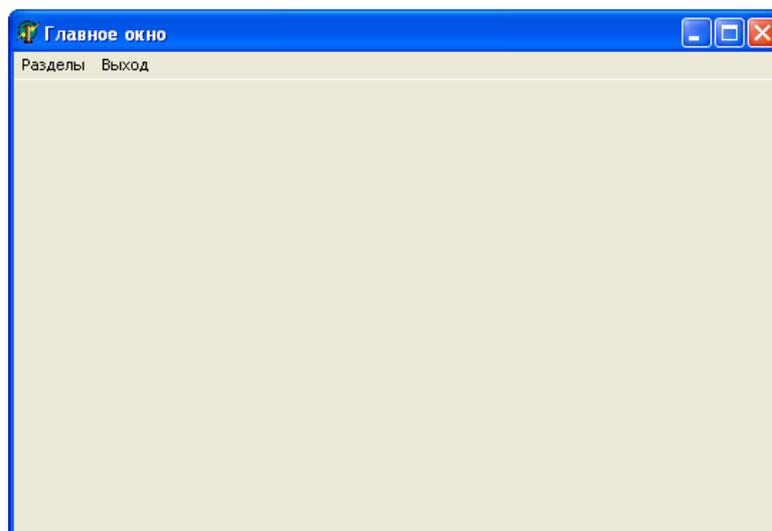


Рисунок 2.1 – Главное окно приложения.

Пункт меню РАЗДЕЛЫ содержит четыре пункта:

- Работа с изображениями;
- Текстовый редактор;
- Настройки;
- Построение графиков.

2. При выборе пункта РАБОТА С ИЗОБРАЖЕНИЯМИ появляется диалоговое окно выбора графического файла стандартного вида. В случае выбора файла появляется форма с выбранным изображением, в строке состояния отображается имя открытого файла и координаты мыши (рисунок 2.2).



Рисунок 2.2 – Форма работы с изображением

3. При выборе пункта ТЕКСТОВЫЙ РЕДАКТОР выводится форма, содержащая многострочное окно редактирования Мемо и текстовый редактор RichEdit.

С визуальными компонентами Мемо и RichEdit необходимо связать всплывающую строку подсказки и всплывающее меню для организации работы с текстом (копировать, перемещать, изменять шрифт). Примерный вид формы приведен на рисунке 2.3.

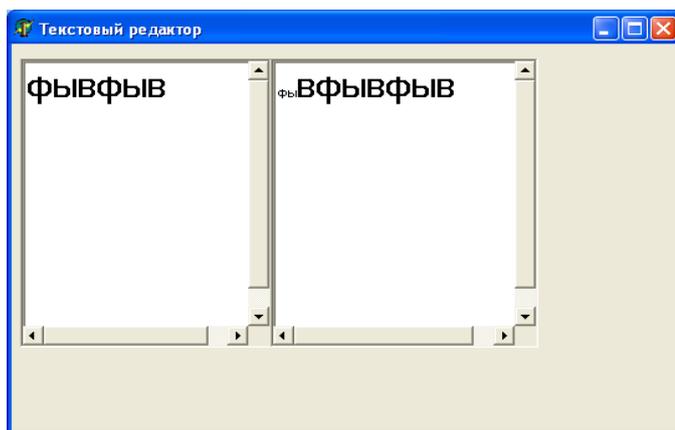


Рисунок 2.3– Текстовый редактор

4. Форма НАСТРОЙКИ представляет собой многостраничное окно. Окно имеет три страницы:

- Ввод чисел и дат (рисунок 2.4);

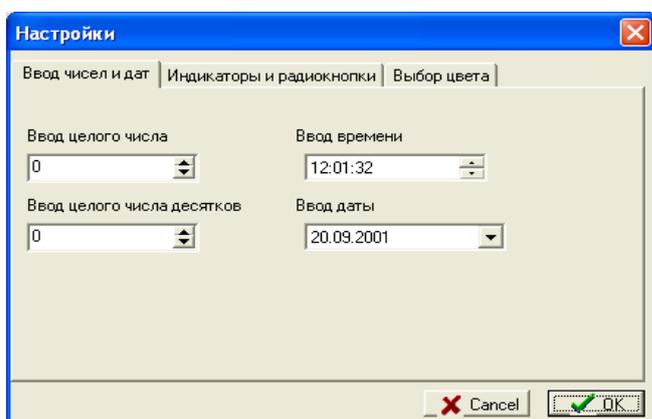


Рисунок 2.4 – Страница «Ввод чисел и дат»

5. Индикаторы и радиокнопки (рисунок 2.5);

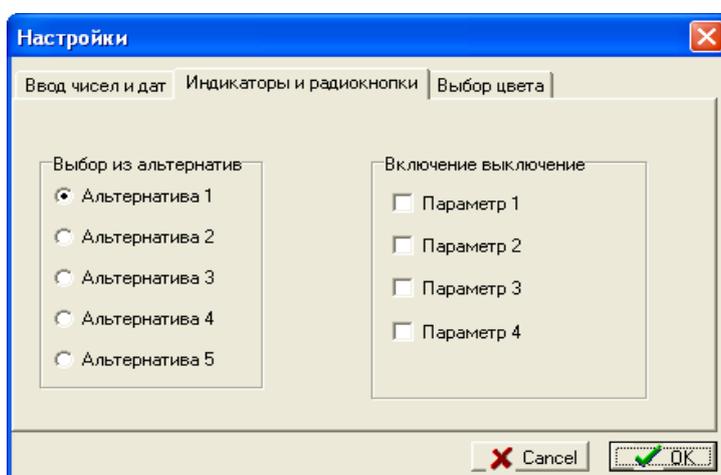


Рисунок 2.5 – Страница «Индикаторы и радиокнопки»

## 6. Выбор цвета (рисунок 2.6).

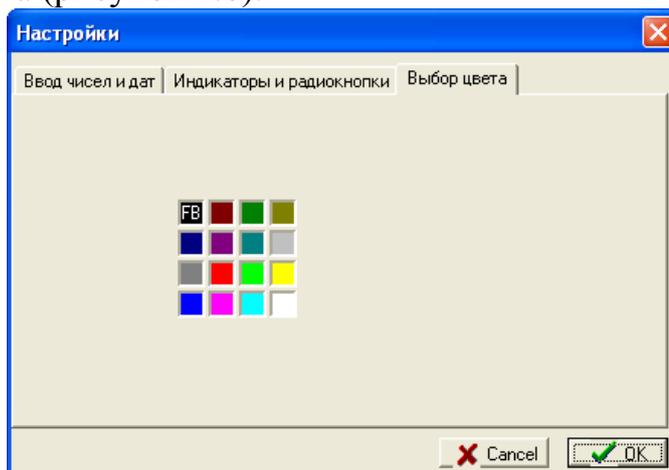


Рисунок 2.6 – Страница «Выбор цвета»

## 7. При нажатии кнопки ОК необходимо:

1. запросить, необходимо ли сохранить настройки (окно сообщения Windows)

2. организовать сохранение установленных настроек в текстовом файле. Для выбора имени файла вызывается диалоговое окно стандартного вида.

8. При выборе пункта ПОСТОЕНИЕ ГРАФИКОВ выводится форма с графиками функций  $y=\sin(x)$  и  $y=\cos(x)$  (рисунок 2.7). При изменении размеров окна график должен масштабироваться.

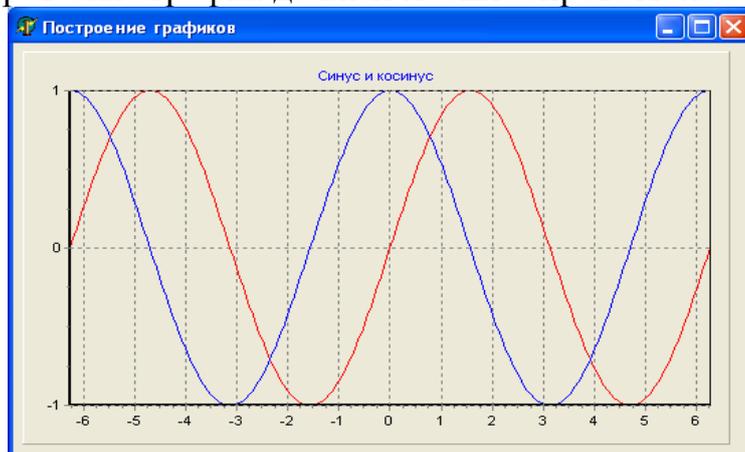


Рисунок 2.7 – Форма ПОСТРОЕНИЕ ГРАФИКОВ

## Критерии оценки:

- оценка «отлично» выставляется студенту, если задание выполнено верно и в полном объеме;
- оценка «хорошо» выставляется студенту, если задание выполнено с

незначительными замечаниями;

- оценка «удовлетворительно» выставляется, если задание выполнено на базовом уровне, но с ошибками;

- оценка «неудовлетворительно» выставляется, если содержится большое количество ошибок, задание не выполнено.

## Текущая аттестация 2

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ  
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»  
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

### Комплект заданий для проведения текущей аттестации №2 (в форме контрольной работы)

#### Пример варианта контрольной работы № 1

1. *Составить линейный алгоритм:* расстояние между автомобилями —  $S$  км. Определить расстояние между ними через  $t$  ч, если первый движется со скоростью  $v_1$ , а второй —  $v_2$  км/ч. Примечание. Рассмотреть случаи движения навстречу друг другу и в противоположные стороны друг от друга.

2. *Составить алгоритм с развилкой:* выяснить, что больше: площадь правильного шестиугольника со стороной  $a$  или прямоугольника с измерениями  $b, c$ .

3. *Составить алгоритм с использованием цикла:* найти сумму членов ряда, для которых выполняется неравенство  $|a_i| \geq \square$ :

$$a_i = \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

4. *Составить алгоритм решения задачи целочисленной арифметики:* вычеркнуть из записи данного натурального числа  $N$  первую цифру.

5. *Составить алгоритм решения задачи с использованием вспомогательных алгоритмов:* поменять порядок следования цифр в заданном натуральном числе  $N$  на обратный, удалив при этом из записи получаемого числа все вхождения цифры  $k$ .

#### Пример варианта контрольной работы №2

1. Изобразить на плоскости  $(X, Y)$  область, в которой и только в которой истинно указанное выражение

$(X \leq 0) \text{ and } (\text{Sqr}(X) + \text{Sqr}(Y) \leq 4) \text{ or } (X \geq 0) \text{ and } (Y \leq -X + 2) \text{ and } (Y \geq X - 2)$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Написать тот вариант программы, который, по Вашему мнению, будет правильным.

```
Program My_Error;  
Var X, Y, Сумма: Integer;  
Begin
```

```

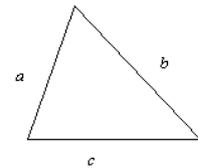
WriteLn( Введите значение X); ReadLn(X);
WriteLn( Введите значение Y); ReadLn(Y);
Сумма = X + Y;
WriteLn(Сумма)
End.

```

3. Составить программу идентификации треугольника по сторонам  $a$ ,  $b$ ,  $c$ . Определяемое свойство: является прямоугольным или не является прямоугольным (удобно использовать сравнение квадрата одной из сторон с суммой квадратов двух других сторон).

4. Решить задачу с использованием оператора выбора: по введенному числу грибов  $k$  напечатать фразу “Мы нашли в лесу  $k$  грибов” и согласовать при этом окончание слова “гриб” с числом  $k$ . (Количество грибов может быть любым целым числом: 1, 3, 34, 127 и т.п. Окончание фразы определяется значением последней цифры.)

5. Точно и четко сформулировать условие задачи, которая решается в данной программе:



```

Program Pr;
Var K, I, D, S: Integer;
Begin WriteLn(‘Введите значение K: ’); ReadLn(K);
For I := 100 To 300 Do
Begin D := 2;
S := 0;
While D <= I div 2 + 1 Do
Begin If I mod D = 0 Then S := S + D;
D := D + 1
End;
If S = K Then WriteLn(I);
End
End.

```

6. Решить задачу с использованием цикла: найти  $k$ -е простое число в арифметической прогрессии 11, 21, 31, 41, 51, 61, ...

7. Определить, что будет напечатано программой по ходу ее выполнения:

```

Program Schet;
Var M, N: Word;
Begin M := 0; N := 0;
Repeat M := M + 1;
N := N + M;
Write(M:5, N:3)
Until N mod 7 = 0;
Writeln;
M := 0; N := 0;
While M + N < 20 Do
Begin M := M + 1; N := N + M; Write(M:5, N:3) End
End.

```

8. Решить задачу с использованием подпрограммы: вывести наибольшую из первых цифр трех заданных чисел. Например, если  $a = 25$ ,  $b = 730$ ,  $c = 1995$ , то надо вывести цифру 7.

9. В данной последовательности  $a_1, a_2, \dots, a_n$  определить максимальное число среди элементов с номерами, кратными числу  $k$ . Например, для последовательности  $-1, 0, 12, -77, 22, -6, 70, 11, 3$  и  $k = 3$  получаем ответ 12.

10. Вычислить

$$\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}$$

( $n$  слагаемых).

### Примерный вариант контрольной работы №3 Массивы, строки

1. Определить, какой объем оперативной памяти требуется для хранения каждой из объявленных переменных. Указать, какую из переменных не удастся разместить в памяти из-за превышения допустимого объема для статических переменных (возникнет ошибка Structure too large).

- Var A : Array[0..99] Of LongInt;
- Var B : Array[Integer] Of Real;
- Var C : Array[-100..100] Of Real;
- Var D : Array['a'..'z'] Of Extended;
- Var E : Array[1..100, 0..39] Of ShortInt;

2. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```
Program Kr_N_4;
Const NMax = 50; Type Mass = Array[0..NMax] Of Real;
Var X : Mass; I, N : Byte; Vsp : Real;
Begin Write('Сколько элементов в массиве? '); ReadLn(N); N := N - 1;
Randomize;
For I := 0 To N Do
Begin X[i] := -50 + Random * 100.1; Write(X[i]:10:5) End;
WriteLn; I := 0;
While I < N Do
Begin Vsp := X[i]; X[i] := X[i + 1];
X[i + 1] := Vsp; I := I + 2
End;
For I := 0 To N Do Write(X[i]:10:5); WriteLn;
End.
```

3. Решить задачу: подсчитать количество элементов массива, каждый из которых не меньше всех своих предшественников (т.е. элементов с меньшими индексами). Например, для массива  $-2, 3, 0, 13, 13, 4, -3$  ответ будет 4.

4. Решить задачу: заполнить квадратную матрицу порядка  $N$  по следующему правилу: элементы выше главной диагонали совпадают с разностью первого и второго индексов, выше побочной диагонали — с произведением индексов. Примечание. К элементам, попадающим в

пересечение областей, применить следующее правило: они получаются, как сумма значений, полученных по первому и второму правилу. Все остальные элементы матрицы равны нулю.

5. *Решить задачу:* в целочисленной прямоугольной таблице размером  $M \times N$  определить наибольшее число, кратное заданному  $K$ , или сообщить об отсутствии таких чисел.

6. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```
Program Kr_4_z2;
Var S : String; M, I, N, J : Byte; Vsp : Char;
Begin
Write('Введите строку: '); ReadLn(S); N := Length(S);
For I := 1 To N - 1 Do
Begin m := I;
For J := I + 1 To N Do If S[j] < S[m] Then m := j;
Vsp := S[i]; S[i] := S[m]; S[m] := Vsp
End;
WriteLn('Ответ: ', S)
End.
```

6. *Решить задачу:* в строке заменить указанный символ  $C$  на символ  $C1$ . Подсчитать число замен.

### **Примерный вариант контрольной работы №4 (рекурсия, множества, записи)**

1. Определить, какой минимальный объём оперативной памяти требуется для хранения каждой из объявленных переменных.

- a) Type CompCh = Record re, im: Real end;  
Var Spis : array[1..100,0..39] of CompCh;
- b) Type Kol = Record name : string[10]; fone : LongInt end;  
Var T: array [-1..10] of Kol;
- c) Var Uch: Record fam, name: string[15]; year : Word end;
- d) Var M: set of '0'..'9';

2. Описать рекурсивную функцию  $pow(x, n)$  от вещественного  $x$  ( $x \neq 0$ ) и целого  $n$ , которая вычисляет величину  $x^n$  согласно формуле

$$x^n = \begin{cases} 1, & x = 0; \\ \frac{1}{x^{|n|}}, & x < 0; \\ x \cdot x^{n-1}, & x > 0. \end{cases}$$

3. Вычислить значения выражений

- a) [2,3,5,7]<=[1..9]; b) ['c', 'b']=['b'..'c'];
- c) trunc(3.9) in [1,3,5]; d) [2..13]\*[3,13..60]+[4..10]-[5..15]\*[6];
- e) [2..10]-[4,6]-[2..12]\*[8..15]; f) (('0'..'7')+['2'..'9'])\*(['a']+['z']).

4. Решить задачу с использованием множества: имеется строка, содержащая русский текст. Вывести те буквы, которые являются общими для первых трёх предложений текста.

5. Заданы некоторые типы данных и объявлены переменные:

```
Type Complex = Record re, im: Real end;
```

```
Point = Record x, y: Real end;
```

```
Var z, w: Complex; p: Point; re: Real;
```

Определить, какие значения будут иметь переменные  $z.re$ ,  $z.im$ ,  $w.re$ ,  $w.im$ ,  $p.x$ ,  $p.y$ ,  $re$  после выполнения следующих операторов:

```
with z do begin re := 0; im := 1 end;
```

```
w := z; re := 2;
```

```
with z do re := 1;
```

```
with z, w do im := -im;
```

```
with p do begin x := re; y := 2 end;
```

6. Сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в массиве:

```
Type Stud = Record
```

```
fam : string[12];
```

```
num : 121..126;
```

```
otm1, otm2, otm3 : 2..5;
```

```
End;
```

```
Var Kurs : array [1..100] of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости

### **Примерный вариант контрольной работы №5 (файлы, модули)**

1. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```
Program Example_File_1;
```

```
Var F : File Of Real; G : File Of Integer;
```

```
S : String; A : Real; B : Integer;
```

```
Begin
```

```
Write('Имя файла? '); ReadLn(S); Assign(F, S);
```

```
Write('Имя файла? '); ReadLn(S); Assign(G, S);
```

```
ReSet(F); ReWrite(G);
```

```
While Not Eof(F) Do
```

```
Begin
```

```
Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)
```

```
End; Close(F); Close(G)
```

```
End.
```

2. Решить задачу: заполнить файл целыми числами из отрезка  $[-1000; 1000]$  с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

3. *Решить задачу:* сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

```
Type Stud = Record  
fam:string[12];  
num:121..126;  
otm1, otm2, otm3: 2..5;  
end;  
Var Kurs: File of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости их студентов.

4. *Решить задачу:* удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

5. *Решить задачу:* разработать модуль для работы с векторами вида  $a(a_1, a_2)$ . Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного  $K$ .

### **Примерный вариант контрольной работы №6 (файлы, модули)**

1. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```
Program Example_File_1;  
Var F : File Of Real; G : File Of Integer;  
S : String; A : Real; B : Integer;  
Begin  
Write('Имя файла? '); ReadLn(S); Assign(F, S);  
Write('Имя файла? '); ReadLn(S); Assign(G, S);  
ReSet(F); ReWrite(G);  
While Not Eof(F) Do  
Begin  
Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)  
End; Close(F); Close(G)  
End.
```

2. *Решить задачу:* заполнить файл целыми числами из отрезка  $[-1000; 1000]$  с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

3. *Решить задачу:* сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

```
Type Stud = Record  
fam:string[12];  
num:121..126;
```

otm1, otm2, otm3: 2..5;  
end;

Var Kurs: File of Stud;

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости их студентов.

4. *Решить задачу:* удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

5. *Решить задачу:* разработать модуль для работы с векторами вида  $a(a_1, a_2)$ . Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного  $K$ .

### **Критерии оценки:**

- оценка «отлично» выставляется студенту, если контрольная работа выполнена верно и в полном объеме;

- оценка «хорошо» выставляется студенту, если контрольная работа выполнена с незначительными замечаниями;

- оценка «удовлетворительно» выставляется, если контрольная работа выполнена на базовом уровне, но с ошибками;

- оценка «неудовлетворительно» выставляется, если содержится большое количество ошибок, контрольная работа не выполнена.