

**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«КОЛЛЕДЖ «КРАСНОСЕЛЬСКИЙ»**

**РАССМОТРЕНО И ПРИНЯТО**

на заседании Педагогического Совета  
СПб ГБПОУ «Колледж «Красносельский»

Протокол №   6   от   07.06.   2024 г.

**УТВЕРЖДАЮ**

Директор СПб ГБПОУ  
«Колледж «Красносельский»

\_\_\_\_\_ Г.И. Софина  
«  \_\_\_  » \_\_\_\_\_ 2024 г.  
Приказ № 101-осн. от   07.06.   2024 г

**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ  
ПО ПРОФЕССИОНАЛЬНОМУ МОДУЛЮ**

ПМ.01 Разработка модулей программного обеспечения для компьютерных систем

**программы подготовки специалистов среднего звена  
по специальности**

09.02.07 Информационные системы и программирование  
(Программист)

**СОГЛАСОВАНО:**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Санкт-Петербург  
2024 г.

**РАССМОТРЕНО И ОДОБРЕНО**

На заседании МК СПб ГБПОУ «Колледж «Красносельский»

Протокол № \_\_\_\_\_ от \_\_\_\_\_ 2024 г.

Председатель МК \_\_\_\_\_ Н.В. Медведева

**Организация-разработчик:** СПб ГБПОУ «Колледж «Красносельский»

## СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА	4
2. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ	6
МДК 01.01 Разработка программных модулей	6
МДК 01.02 Поддержка и тестирование программных модулей	16
Примерные тестовые задания	16
МДК 01.03 Разработка мобильных приложений	25
Проверочные тесты	25
3. МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ	28
МДК 01.01 Разработка программных модулей	28
Вопросы к дифференцированному зачету	28
МДК 01.02 Поддержка и тестирование программных модулей	29
Вопросы к дифференцированному зачету	29
МДК 01.03 Разработка мобильных приложений	30
Примеры практико-ориентированных заданий	30
Вопросы к дифференцированному зачету	32
МДК 01.04 Системное программирование	32
Примерные практико-ориентированные задания	32
Примерные билеты к дифференцированному зачету	33
ПМ.04 Разработка модулей программного обеспечения для компьютерных систем	38
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ	39

## 1. ОБЩАЯ ХАРАКТЕРИСТИКА

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся.

КОС включают контрольные материалы для проведения текущего контроля и промежуточной аттестации.

Результатом освоения междисциплинарного курса является формирование у обучающихся следующих общих и профессиональных компетенций:

ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности
ОК 3	Планировать и реализовывать собственное профессиональное и личностное развитие
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста
ОК 6	Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей
ОК 7	Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях
ОК 8	Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности
ОК 9	Пользоваться профессиональной документацией на государственном и иностранном языке
ПК 1.1	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием
ПК 1.2	Разрабатывать программные модули в соответствии с техническим заданием
ПК 1.3	Выполнять отладку программных модулей с использованием специализированных программных средств
ПК 1.4	Выполнять тестирование программных модулей
ПК 1.5	Осуществлять рефакторинг и оптимизацию программного кода
ПК 1.6	Разрабатывать модули программного обеспечения для мобильных платформ

В результате освоения междисциплинарного курса обучающийся должен иметь:

Практический опыт	Умения	Знания
<ul style="list-style-type: none"> <li>- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;</li> <li>- в использовании инструментальных средств на этапе отладки программного продукта;</li> <li>- в проведении тестирования программного модуля по определенному сценарию;</li> <li>- в разработке мобильных приложений</li> </ul>	<ul style="list-style-type: none"> <li>- осуществлять разработку кода программного модуля на языках низкого и высокого уровней;</li> <li>- создавать программу по разработанному алгоритму как отдельный модуль;</li> <li>- выполнять отладку и тестирование программы на уровне модуля;</li> <li>- осуществлять разработку кода программного модуля на современных языках программирования;</li> <li>- уметь выполнять оптимизацию и рефакторинг программного кода;</li> <li>- оформлять документацию на программные средства</li> </ul>	<ul style="list-style-type: none"> <li>- основные этапы разработки программного обеспечения;</li> <li>- основные принципы технологии структурного и объектно-ориентированного программирования;</li> <li>- способы оптимизации и приемы рефакторинга;</li> <li>- основные принципы отладки и тестирования программных продуктов</li> </ul>

## 2. МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

### МДК 01.01 Разработка программных модулей

Текущий контроль осуществляется в форме тестирования.

**1** Независимость программных продуктов от технического комплекса системы обработки данных, операционной среды, сетевой технологии обработки данных, специфики предметной области и т.п., означает их ...

а) модифицируемость

б) надежность

**в) мобильность**

г) эффективность

**2** Бесбойность и устойчивость в работе программных продуктов, точность выполнения предписанных функций обработки, возможность диагностики возникающих в процессе работы программ ошибок, означает их ...

а) модифицируемость

**б) надежность**

в) мобильность

г) эффективность

**3** Возможность применения алгоритма к целому классу однотипных задач, различающихся конкретным значением, является одним из основных свойств алгоритма ... а) детерминированностью

б) дискретностью

в) определенностью

**г) массовостью**

**4** Максимально возможная интеграция программных продуктов с другими программами, обеспечение обмена данными в общих форматах представления (экспорт/импорт баз данных, внедрение или связывание объектов обработки и др.) означает их ... а) модифицируемость

б) эффективность

в) мобильность

**г) коммуникативность**

**5** Способность программных продуктов к внесению изменений, например расширение функций обработки, переход на другую техническую базу обработки и т.п., означает их ...

**а) модифицируемость**

б) эффективность

в) мобильность

г) учет человеческого фактора

**6** Наличие дружественного интерфейса, контекстно-зависимой подсказки, хорошей документации является таким показателем качества программного продукта, как ...

а) коммуникативность

б) эффективность

в) мобильность

**г) учет человеческого фактора**

**7** В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- постановки задачи,

- выбора критериев эффективности,

- проведения предварительных научно-исследовательских работ,

- разработки технического задания, определяет содержание этапа модели ЖЦ ...

а) Эскизный проект

б) Технический проект

**в) Техническое задание**

г) Рабочий проект

**8** В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- структуры входных и выходных данных,
- уточнения методов решения,
- общего алгоритма,
- разработки документации эскизного проекта, определяет содержание этапа модели

ЖЦ ... **а) Эскизный проект**

- б) Технический проект
- в) Техническое задание
- г) Рабочий проект

**9** В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- уточнения структуры входных и выходных данных,
- разработки алгоритмов,
- форм данных,
- семантики и синтаксиса языка,
- структуры программы,
- конфигурации технических средств,
- плана работ,

определяет содержание этапа модели ЖЦ ...

а) Эскизный проект

**б) Технический проект**

- в) Техническое задание
- г) Рабочий проект

**10** В соответствии с ГОСТ 19.102-77 детальное описание действий в виде:

- программирования и отладки;
- разработки документов;
- подготовки и проведения испытаний;
- корректировки программы и документов по итогам испытаний. определяет содержание

этапа модели ЖЦ ...

а) Эскизный проект

б) Внедрение

в) Техническое задание

**г) Рабочий проект**

**11** Наличие обратных связей между этапами жизненного цикла, определяющих возможность проведения проверок и корректировок проектируемого ПС на каждой стадии разработки, является основной особенностью ...

а) каскадной модели жизненного цикла

**б) итерационной модели жизненного цикла**

в) спиральной модели жизненного цикла ПО

г) модели жизненного цикла UML

**12** Основными особенностями:

- последовательным выполнением входящих в состав ЖЦ этапов,
- окончанием каждого предыдущего этапа до начала последующего,
- отсутствием временного перекрытия этапов,
- отсутствием возврата к предыдущим этапам, - наличием результата только в конце

разработки, характеризуется ...

**а) каскадная модель жизненного цикла**

б) итерационная модель жизненного цикла

в) спиральная модель жизненного цикла ПО

г) модель жизненного цикла UML

**13** Выявлением и устранением ошибок только на стадии тестирования, которая может растянуться во времени или вообще никогда не завершиться, характеризуется ...

**а) каскадная модель жизненного цикла**

- б) итерационная модель жизненного цикла
- в) спиральная модель жизненного цикла ПО
- г) модель жизненного цикла UML

15 Функционирование программного продукта в соответствии с техническим заданием соответствует характеристике эксплуатационных требований ...

**а) правильность**

- б) универсальность
- в) надежность
- г) адаптируемость

16 Обеспечение правильной работы при любых допустимых данных и защиты от неправильных данных соответствует характеристике эксплуатационных требований ... а) правильность

**б) универсальность**

- в) надежность
- г) защищенность

17 Обеспечение правильности результатов при наличии различного рода сбоев соответствует характеристике эксплуатационных требований ...

- а) правильность
- б) универсальность

**в) надежность**

- г) защищенность

18 Обеспечение погрешности результатов не выше заданной соответствует характеристике эксплуатационных требований ...

- а) правильность
- б) проверяемость

**в) точность результатов**

- г) защищенность

19 Использование минимально возможного количества ресурсов технических средств (например, времени микропроцессора, объема оперативной памяти, объема внешней памяти, количества внешних устройств и др.) соответствует характеристике эксплуатационных требований ...

- а) универсальность
- б) надежность
- в) аппаратная совместимость

**г) эффективность**

20 Возможность быстрой модификации с целью приспособления к изменяющимся условиям функционирования соответствует характеристике эксплуатационных требований ...

- а) универсальность
- б) эффективность

**в) адаптируемость**

- г) рентабельность

21 Возможность «параллельного» использования несколькими процессами соответствует характеристике эксплуатационных требований ...

- а) универсальность
- б) программная совместимость
- в) повторная входимость

**г) рентабельность**

22 Функциональная диаграмма имеет аббревиатуру ... а) ERD

**б) SADT**

- в) DFD
- г) STD

**23** Диаграмма потоков данных имеет аббревиатуру ... а) ERD

б) SADT

в) **DFD**

г) STD

**24** Диаграмма переходов состояний имеет аббревиатуру ... а) ERD

б) SADT

в) DFD

г) **STD**

**25** Диаграмма «сущность—связь» имеет аббревиатуру ... а) **ERD**

б) SADT

в) DFD

г) STD

**26** Вершиной древовидной структуры диаграмм, представляющей собой самое общее описание системы и ее взаимодействия с внешней средой, является следующий вид диаграммы IDEF0 ...

а) диаграмма декомпозиции

б) диаграмма дерева узлов

в) **контекстная диаграмма**

г) диаграмма для экспозиции

**27** Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования **UML** объединяет несколько моделей. Содержит описание функций программного обеспечения с точки зрения пользователя ...

а) **модель использования**

б) логическая модель

в) модель реализации

г) модель развертывания

**28** Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования **UML** объединяет несколько моделей. Описывает ключевые понятия моделируемого программного обеспечения (классы, интерфейсы и т. п.), т. е. средства, обеспечивающие его функциональность ...

а) **логическая модель**

б) модель реализации

в) модель процессов

г) модель развертывания

**29** Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования **UML** объединяет несколько моделей. Определяет реальную организацию программных модулей в среде разработки ...

а) модель использования

б) **модель реализации**

в) модель процессов

г) модель развертывания

**30** Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования **UML** объединяет несколько моделей. Отображает организацию вычислений и позволяет оценить производительность, масштабируемость и надежность программного обеспечения ...

а) модель использования

б) модель реализации

в) **модель процессов**

г) модель развертывания

**31** Спецификация разрабатываемого программного обеспечения при использовании языка визуального моделирования **UML** объединяет несколько моделей. Показывает, каким образом программные компоненты размещаются на конкретном оборудовании ...

а) модель использования

б) модель реализации

в) модель процессов

г) **модель развертывания**

**32** Служит связующим звеном между процессами, т. е. позволяет описать порядок взаимодействия процессов ...

а) **нотация**

б) прецедент

в) поток событий

г) сценарий

**33** Подробное процедурное описание вариантов использования системы всеми заинтересованными лицами, а также внешними системами, реализует ... а) нотация

б) **прецедент**

в) поток событий

г) сценарий

**37** Идентификатором, сформированным по всем правилам языка C, можно назвать выражение ...

а) lizonьka

б) **lizo4ka\_**

в) no\_naroshky

г) egoza-liza

**38** Для описания основных (стандартных) типов данных на языке C определены следующие ключевые слова ...

а) int, char, bool, double, void

б) int, char, float, double, bool

в) int, wchar\_t, float, double, char

г) **int, char, float, double, void**

**39** Символьная константа на языке C может быть представлена в виде ... а) 'Q1'

б) '<='

в) ';'

г) "z"

**40** Десятичная константа на языке C может быть представлена в виде ... а) 020

б) **101**

в) 01

г) 0x20

**1** Верным на языке C является следующее объявление и инициализация переменных

...

а) **double n, k=3;**

б) float n, k, int z=-1;

в) char a, b='No';

г) float \*t=4;

**2** Строковой константой на языке C считается следующее выражение... а) a"l"v"f"

б) **"Mus%tika!"**

в) 'Slonik'

г) "Slovo\_no"1"1

**3** Идентификатором не является ...

а) имя объекта

б) название свойства

в) **зарезервированное слово**

г) имя поля в структуре

**4** Верным на языке C является следующее объявление и инициализация переменных

...

а) short char k, z='б';

б) *float k, \*m=&k;*

в) short signed int c, m=1.0;

г) long int my='0', 8c=3;

5 Наполнить ячейку памяти значением призван оператор ...

а) неполного ветвления

б) полного ветвления

в) *присваивания*

г) цикла

6 Фрагмент кода программы на языке C char \*s, c; int i=0, j=0;

...

```
while (s[i]){ if (s[i]!=c) s[j++]=s[i]; i++; } s[j]='\0';
```

...

а) меняет местами символы в строке

б) заменяет символ на нулевой байт

в) вставляет символ, если такого нет в строке

г) *удаляет символ из строки*

7 Функция на языке C void func1(int \*a, int k) { int i, j, m;

```
for (i = 0; i < k; i++) a[i]=i;
```

```
m=2;
```

```
while (m<k) { if (a[m]!=0) { j=m*2; while (j<k){ a[j]=0; j=j+m;
```

```
}
```

```
}
```

```
m=m+1;
```

```
}
```

```
} возвращает k-элементный массив, в котором кроме нулей содержатся... а) четные
```

числа

б) нечетные числа

в) *простые числа*

г) четные удвоенные числа

8 Функция на языке C void func(int \*\*a, int n, int m) { int i, j, k, n1;

```
if (n%2==0) n1=n/2;
```

```
else n1=n/2+1; printf("n1 = %d", n1); for (j = 0; j < m; j++) { k = 0; for (i = n-1; i >=n1; i-
```

```
-) { a[i][j] = a[i][j] + a[k][j]; a[k][j] = a[i][j] - a[k][j];
```

```
a[i][j] = a[i][j] - a[k][j];
```

```
k+=1;
```

```
}
```

```
}
```

```
} возвращает двумерный массив, в котором ...
```

а) изменен порядок следования элементов каждой строки на обратный

б) изменен порядок следования двух рядом стоящих элементов массива

в) *изменен порядок следования элементов каждого столбца на обратный*

г) изменен порядок следования двух рядом стоящих элементов каждого столбца

9 После выполнения фрагмента кода программы char ZNAC= '@';

```
int x=1, y=-1, z=2;
```

```
switch (ZNAC) {
```

```
case '~': x = x + y + z; case '@': z = 2*y - 3*x; case '7': x = 5*y - z; break; case 'a': y = x +
```

```
3*(y - z); case '*': x = y * z; case '#': x = y - x - z; break; default : x = y - 1;
```

```
}
```

переменная x будет иметь значение...

а) -3

б) -16

в) -2

з) 0

**10** После выполнения следующего программного кода

```
int i, j, n = 0, sqrs[4][3] = {1, 8, 2, 4, 3, 9, 4, 6, 5, 2, 6, 7}; for (i = 0; i < 4; i++) for (j = 0; j < 3; j++) if (sqrs[i][j] % 2 == 0 && i % 2 != 0) n += sqrs[i][j];
```

пользователь увидит, что переменная **n** будет содержать ...

а) сумму четных элементов нечетных столбцов двумерного массива

б) сумму четных элементов нечетных строк двумерного массива

**в) сумму четных элементов четных строк двумерного массива**

г) сумму нечетных элементов четных столбцов двумерного массива

**11** Прототип функции

```
char isn(int k, char *r, float *m);
```

позволяет обратиться к функции с помощью фрагмента программы ...

а) `int a = -2; char s[10] = "Virtual&", simb; float n = 6.1;`

`*simb = isn(a, &s, &n);`

б) `int a = -2; char s[10] = "Virtual%", simb; float n = 6.1; simb = isn(a, *s, *n);`

**в) `int a = -2; char s[10] = "Virtual3", simb; float n = 6.1; simb = isn(a, s, &n);`**

г) `int a = -2; char s[10] = "Virtual4", simb; float n = 6.1; simb = isn(a, &s, &n);`

**12** При обращении к функции в соответствии с требованиями хорошего стиля программирования рекомендуется (но не обязательно), чтобы ...

**а) имена формальных и фактических параметров были различными**

б) количество формальных параметров соответствовало количеству фактических параметров

в) типы формальных параметров соответствовали типам фактических параметров

г) порядок перечисления фактических параметров соответствовал порядку перечисления формальных параметров

**13** В объектно-ориентированной среде программирования обработчики любых событий для компонентов размещаются в окне ...

а) Формы

б) Инспектора объектов

**в) Кода**

г) Деревя объектов

**14** В объектно-ориентированной среде программирования к любому компоненту в программе можно обратиться по его свойству ...

а) Caption

б) Text

в) Align

**з) Name**

**15** Запись вещественного числа **aEb** означает ... а) число **a** умножить на **b** в степени 10

б) число **b** умножить на 10 в степени **a**

**в) число **a** умножить на 10 в степени **b****

г) число **b** умножить на **a** в степени 10

**16** Переменные, которые передаются подпрограмме в момент ее вызова из основной программы, называются ...

а) формальными параметрами

б) глобальными переменными

**в) фактическими параметрами**

г) локальными переменными

**17** Переменные, которые определяются в заголовке подпрограммы при её объявлении и описании, называются ...

**а) формальными параметрами**

- б) глобальными переменными
- в) фактическими параметрами
- г) локальными переменными

**18** Получить наименования всех отпущенных со склада товаров можно с помощью SQL-оператора...

- а) `SELECT TOVAR FROM RASHOD`
- б) **`SELECT DISTINCT TOVAR FROM RASHOD`**
- в) `SELECT * FROM RASHOD ORDER BY TOVAR`
- г) `SELECT TOVAR FROM RASHOD WHERE TOVAR = :PARAM1`

**19** Общее количество отпуска по каждому из товаров можно получить с помощью SQL-оператора ...

- а) **`SELECT R.TOVAR, SUM(R.KOLVO) AS OTPUSK FROM RASHOD R GROUP BY R.TOVAR`**
- б) `SELECT DAT_RASH, COUNT(DISTINCT POKUP) FROM RASHOD GROUP BY DAT_RASH`
- в) `SELECT R.TOVAR, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR`
- г) `SELECT R.TOVAR, R.DAT_RASH, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR, R.DAT_RASH`

**20** Общая цена каждого из отпущенных товаров определяется с помощью SQL-оператора ...

- а) **`SELECT R.TOVAR, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR`**
- б) `SELECT R.TOVAR, SUM(R.KOLVO) AS OTPUSK FROM RASHOD R GROUP BY R.TOVAR`
- в) `SELECT DAT_RASH, COUNT(DISTINCT POKUP) FROM RASHOD GROUP BY DAT_RASH`
- г) `SELECT R.TOVAR, R.DAT_RASH, SUM(R.KOLVO * T.ZENA) FROM RASHOD R, TOVARY T WHERE T.TOVAR = R.TOVAR GROUP BY R.TOVAR, R.DAT_RASH`

**21** SQL-оператор

**`SELECT TOV, SUM(KOLVO) AS OTPUSK_5000 FROM RASHOD GROUP BY TOV HAVING MIN(KOLVO)>=5000;`** определяет ...

а) сумму отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не более 5000 единиц

б) *общее количество отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не менее 5000 единиц.*

в) сумму отпущенного товара по каждому виду товара для тех покупок, в которых минимальное количество покупаемого товара не менее 5000 единиц

г) количество отпущенного товара для тех покупок, в которых максимальное количество покупаемого товара не менее 5000 единиц

**22** SQL-оператор

**`SELECT R.TOV, SUM(R.KOLVO*T.ZENA) AS STOIM FROM RASHOD R, TOVARY T`**

**`WHERE R.TOV = T.TOVAR GROUP BY R.TOV HAVING COUNT(R.TOV)>3;`**

определяет ...

а) *общую стоимость каждого из отпущенных товаров, количество отпуска которого более 3-х раз*

б) сумму отпущенного товара по каждому виду товара, количество отпуска которого более 3-х раз

в) количество отпущенного товара по каждому виду товара, количество отпуска которого не более 3-х раз

г) общую стоимость каждого из отпущенных товаров, количество отпуска которого не более 3-х раз

**23** Абстрактным называется класс, который ... а) не содержит функций

б) не содержит переменных

в) содержит хотя бы одну виртуальную функцию

*г) содержит хотя бы одну чисто виртуальную функцию*

**24** Код программы на языке C++

```
template <class X> void swapargs(X &a, X &b){ X temp; temp = a; a = b; b = temp;
}
```

содержит ...

*а) обобщенную функцию, меняющую местами две переменные*

б) шаблон, обнуляющий две переменные

в) виртуальную функцию, меняющую местами две переменные

г) функцию, обнуляющую две переменные

**25** Код

```
Query1->Close();
```

```
Query1->SQL->Clear();
```

```
Query1->SQL->Add ("select TOVAR, ED_IZM, ZENA");
```

```
Query1->SQL->Add ("from TOVARY");
```

```
Query1->SQL->Add ("order by TOVAR");
```

```
Query1->Open(); определяет запрос ... а) статический
```

*б) динамический*

в) статический параметрический

г) динамический параметрический

**26** Код

```
Query1->Close();
```

```
Query1->SQL->Clear();
```

```
Query1->SQL->Add ("select TOVAR, ED_IZM, ZENA");
```

```
Query1->SQL->Add ("from TOVARY");
```

```
Query1->SQL->Add ("where TOVAR = :TOVAR ");
```

```
Query1->Open(); определяет запрос ... а) статический
```

б) динамический

в) статический параметрический

*г) динамический параметрический*

**27** Модульное тестирование предполагает ...

*а) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции*

б) поиск каких-либо проблем в интерфейсах и взаимодействиях между компонентами программы

в) тестирование интегрированной системы на ее соответствие исходным требованиям

г) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

**28** Интеграционное тестирование предполагает ...

а) тестирование минимально возможного для тестирования компонента, например отдельного класса или функции

*б) поиск каких-либо проблем в интерфейсах и взаимодействиях между компонентами программы*

в) тестирование интегрированной системы на ее соответствие исходным требованиям

ям

г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**29** Системное тестирование предполагает ...

а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

**б) тестирование интегрированной системы на ее соответствие исходным требованиям**

в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**30** Альфа-тестирование предполагает ...

а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

б) тестирование интегрированной системы на ее соответствие исходным требованиям

**в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика**

г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок

**31** Бета-тестирование предполагает ...

а) поиск каких-либо проблем в интерфейсах и взаимодействии между компонентами программы

б) тестирование интегрированной системы на ее соответствие исходным требованиям

в) имитацию реальной работы с системой штатными разработчиками либо реальной работы с системой потенциальными пользователями/заказчиком на стороне разработчика

**г) распространение версии с ограничениями (по функциональности или времени работы) для некоторой группы лиц с тем, чтобы убедиться, что продукт содержит достаточно мало ошибок**

**32** Каждая команда программы должна быть выполнена хотя бы один раз в соответствии с методом тестирования ...

**а) покрытие операторов;**

б) покрытие решений;

в) покрытие условий;

г) покрытие решений/условий;

**33** Каждое направление перехода в программе должно быть реализовано, по крайней мере, один раз в соответствии с методом тестирования ...

а) покрытие операторов;

**б) покрытие решений;**

в) покрытие условий;

г) покрытие решений/условий;

**34** Все возможные результаты каждого условия в решении должны быть выполнены, по крайней мере, один раз в соответствии с методом тестирования ...

а) покрытие операторов

б) покрытие решений

**в) покрытие условий**

г) покрытие решений/условий

**35** Видимость переменных и методов класса только в пределах данного модуля определяет спецификатора доступа ...

*а) private*

б) protected

в) public

г) published

**36** Видимость переменных и методов класса в классах потомков определяет спецификатора доступа ...

а) private

*б) protected*

в) public

г) published

**37** Видимость переменных и методов класса в любом месте программы определяет спецификатора доступа ...

а) private

б) protected

*в) public*

г) published

**38** Процесс, в ходе которого один объект может приобретать свойства другого, называется ...

*а) наследование*

б) инкапсуляция

в) полиморфизм

г) динамический полиморфизм

**39** Процесс перегрузки функций и операторов реализует принцип ... а) наследования

б) инкапсуляции

*в) статического полиморфизма*

г) динамического полиморфизма

**40** На основе наследования и виртуальных функций реализуется принцип ... а) наследования

б) инкапсуляции

в) статического полиморфизма

*г) динамического полиморфизма*

#### **МДК 01.02 Поддержка и тестирование программных модулей**

##### *Примерные тестовые задания*

1. Delphi – это...

1) Язык программирования низкого уровня

**2) Система объектно-ориентированного визуального программирования**

3) Программа обработки видео

4) Программа для перевода кода программы на язык машинных кодов.

2. RAD – это...

1) Программа записи видео

2) Средство модульного программирования

**3) Среда быстрой разработки приложений**

3. Язык программирования Delphi

**1) Object Pascal**

2) Java

3) C#

4. IDE – это...

1) История развития Delphi

2) Приложение, разрабатываемое программистом

**3) Интегрированная среда разработки**

5. IDE не включает в себя: (несколько вариантов ответа)

- 1) Редактор кода
- 2) Высокопроизводительный компилятор в машинный код
- 3) Объектно-ориентированную модель компонент
- 4) Эффекты и переходы**
- 5) Визуальное построение приложений
- 6) Сопровождение ПП**
- 7) Средство для построения баз данных

6. Объектно-ориентированная программа - ...

- 1) Программирование, основанное на объектах
- 2) Совокупность объектов и способов их взаимодействия**
- 3) Структура среды программирования

7. Установить соответствие

1) Объект	а) Атрибуты (основные характеристики), которые описывают особенности объекта (цвет, ширина, положение и т.д.)
2) Событие	б) Совокупность данных (компонентов) и методов работы с ними
3) Свойство	с) Отклик на внешнее воздействие
<b>1 – б, 2 – с, 3 – а</b>	

8. Дополнить предложение. Основным окном разрабатываемого приложения является

...

- 1) Код
- 2) Форма**
- 3) Библиотека
- 4) Объект

9. Дополнить предложение. Коды для стандартных окон диалога и кнопок в системе Delphi получены от ...

- 1) Компилятора
- 2) C++
- 3) Windows**

10. Названия процедур на определенное событие:

- 1) Компоненты
- 2) Обработчики событий**
- 3) Редактор кода

11. Назначение главного меню:

- 1) Осуществление функций управления при разработке программ**
- 2) Сопровождение программных продуктов
- 3) Автоматизирует процесс тестирования программ

12. Назначение панели инструментов:

- 1) Обработка событий
- 2) Меню команд быстрого доступа к командам, содержащее набор кнопок, функции которых эквивалентны наиболее часто употребляемым командам Главного меню**
- 3) Построение баз данных

13. Назначение палитры компонентов:

- 1) содержит пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение**
- 2) Подделка подписей
- 3) Управление Paint

14. Дополнить предложение. Главное окно управляет окнами ... (несколько вариантов ответа)

- 1) **Инспектор объектов**
  - 2) Язык ассемблера
  - 3) **Редактора кода**
  - 4) **Проектировщика форм**
  - 5) Трансляции программы
15. Заготовка разрабатываемого приложения - ...
- 1) Компилятор
  - 2) Свойство
  - 3) **Окно проектировщика форм**
  - 4) Главное окно
16. Действия которые нельзя выполнить с помощью проектировщика форм
- 1) Добавить компоненты в форму
  - 2) **Сменить язык программирования**
  - 3) Модифицировать форму и её компоненты
  - 4) Связать обработчики событий компонента с программой на Object Pascal, содержащейся в редакторе кода
  - 5) **Документирование программы**
17. Дополнить предложение. Инспектор объектов позволяет ... *(несколько вариантов ответа)*
- 1) **Устанавливать свойства объектов**
  - 2) Изменять структуру программного кода
  - 3) Изменять наследование классов объектов
  - 4) **Назначать методы обработки событий**
18. Дополнить предложение. Инспектор объектов состоит из ... *(несколько вариантов ответа)*
- 1) Unit – программного модуля
  - 2) **Properties – списка свойств,**
  - 3) **Events – списка событий.**
  - 4) Begin...end.
19. Окно содержащее текст программы на языке Object Pascal, связанный с каждой формой в приложении:
- 1) Окно компилятора
  - 2) **Окно редактора кода**
  - 3) Окно проектировщика
  - 4) Главное окно
20. Задание метки текста из редактора ввода:
- 1) Memo1.Caption := Edit1.Text;
  - 2) **Label1. Caption := Edit1.Text;**
  - 3) Form1. Caption := Edit1.Text;
21. Обнуление строки ввода:
- 1) **Edit1.Text :='';**
  - 2) Edit1.Text := TMemo;
  - 3) Edit1.Text :='Закругляемся'
22. Передача фокуса ввода на редактор ввода
- 1) Label1.SetFocus;
  - 2) Edit1.Add;
  - 3) **Edit1.SetFocus;**
23. Открыть файл проекта Project1 можно, нажав:
- 1) CTRL+F4
  - 2) CTRL+ALT+F10
  - 3) **CTRL+F12**
  - 4) ALT+F12

24. Для обозначения комментария не используются:
- 1) (\*комментарий\*)
  - 2) \комментарий/
  - 3) { комментарий }
  - 4) //комментарий
25. Зарезервированное слово, объявляющее блок подключаемых к проекту модулей:
- 1) Program
  - 2) Begin..end
  - 3) **Uses**
  - 4) Forms
26. Дополнить предложение. Объект Application определяет ...
- 1) **Свойства и методы программы-приложения для Windows**
  - 2) Отображение размещения формы
  - 3) Перемещение по программному коду
27. Метод Application.Run:
- 1) Создает окно формы, регистрирует входящие в него компоненты
  - 2) **Подключает автоматический цикл обработки сообщений Windows к приложению**
  - 3) Выполняет подпрограммы раздела инициализации модулей приложения в случае их наличия
28. Контейнер, содержащий все элементы программы:
- 1) Unit1
  - 2) **Form1**
  - 3) Edit1
  - 4) Memo1
29. Дополнить предложение. Свойство **Position** определяет ...
- 1) **Размещение и размер формы**
  - 2) Подключение модулей
  - 3) Цвет шрифта
30. **poDesigned:**
- 1) Форма выводится в центр экрана
  - 2) Windows автоматически определяет начальную позицию и размеры формы
  - 3) **Форма отображается в той позиции и с теми размерами, которые были установлены при её конструировании**
31. Определяет способ выравнивания компонента внутри контейнера:
- 1) Свойство Position
  - 2) **Свойство Align**
  - 3) Свойство Memo
  - 4) Свойство ScrollBars
32. **alClient:**
- 1) **Компонент занимает всю поверхность контейнера**
  - 2) Выравнивание не используется
  - 3) Компонент помещается в нижнюю часть контейнера
33. **alCustom:**
- 1) Компонент помещается в верхнюю часть контейнера
  - 2) Компонент помещается в нижнюю часть контейнера
  - 3) **Размеры и положение компонента устанавливаются разработчиком**
34. Объект **Memo1** - ...
- 1) Однострочный редактор
  - 2) **Многострочный редактор, содержащий несколько строк текста**
  - 3) Текстовая надпись
35. Свойство **ScrollBars:**

- 1) **Задаёт в поле редактирования полосы прокрутки**
  - 2) Определяет способ выравнивания компонента внутри контейнера
  - 3) Задаёт имя объекта
36. Функциональная кнопка с изображением устанавливается из:
- 1) Палитра компонентов -> Standart -> Button
  - 2) Палитра компонентов -> Standart -> Label
  - 3) **Палитра компонентов -> Additional -> BitBtn**
37. Дополнить предложение. За стиль оформления внешней и внутренней рамок отвечают свойства ... (несколько вариантов ответа)
- 1) Align
  - 2) **BevelOuter**
  - 3) **BevelInner**
  - 4) ssBoth
  - 5) Caption

38. Установить соответствие

1) bvLowered	a) Компонент помещается в нижнюю часть контейнера
2) poScreenCenter	b) Есть обе полосы прокрутки
3) bvNone	c) Рамка вдавлена
4) alBottom	d) Форма выводится в центре экрана, её высота и ширина не изменяются
5) ssBoth	e) Рамка отсутствует
<b>1 – c, 2- d, 3 – e, 4 – a, 5 -b</b>	

39. VCL – это...

- 1) Приложение Delphi
  - 2) Алгоритмизация решения задачи
  - 3) **Библиотека визуальных компонентов**
  - 4) Характеристики объектов
40. Дополнить предложение. Компоненты, которые видны на форме, как во время создания приложения, так и во время работы приложения называются ...
- 1) **Визуальными**
  - 2) Реальными
  - 3) Невизуальными
  - 4) Основными
41. К невидимым компонентам относят:
- 1) Кнопки, метки, списки блоков
  - 2) **Таймеры, компоненты для работы с базами данных, списки изображений**
  - 3) Библиотеки, звуки, коды

42. Установить соответствие:

1) Standart	a) Обеспечивает доступ к 32-битным элементам Windows
2) Additional	b) Компоненты, реализующие интерфейс с пользователем и процесс управления данными для БД
3) Win32	c) Включает стандартные компоненты, обеспечивающие некоторые функции интерфейса пользователя
4) System	d) Специализированные компоненты, организующие доступ к БД
5) DataAccess	e) Набор компонентов для доступа к системным ресурсам (OLE, DDE)
6) DataControl	f) Дополнительные интерфейсные компоненты для красочного оформления приложения
<b>1) – c, 2) – f, 3) – a, 4) – e, 5) – d, 6) – b;</b>	

43. Дополнить предложение. Имя компонента состоит из... (несколько вариантов ответа)

- 1) **Названия компонента**
- 2) Заготовки компонента
- 3) **Порядкового номера компонента**
- 4) Произвольного значения ItemIndex

44. Дополнить предложение. Свойства Height и Width показывают...

- 1) Какое действие должен выполнить программист
- 2) **Размеры компонента**
- 3) Исполняемый блок программы

45. Дополнить предложение. Проект Delphi представляет собой...

- 1) **Набор программных единиц – модулей**
- 2) Основной интерфейсный элемент в Delphi
- 3) Атрибуты (основные характеристики), которые описывают особенности объекта

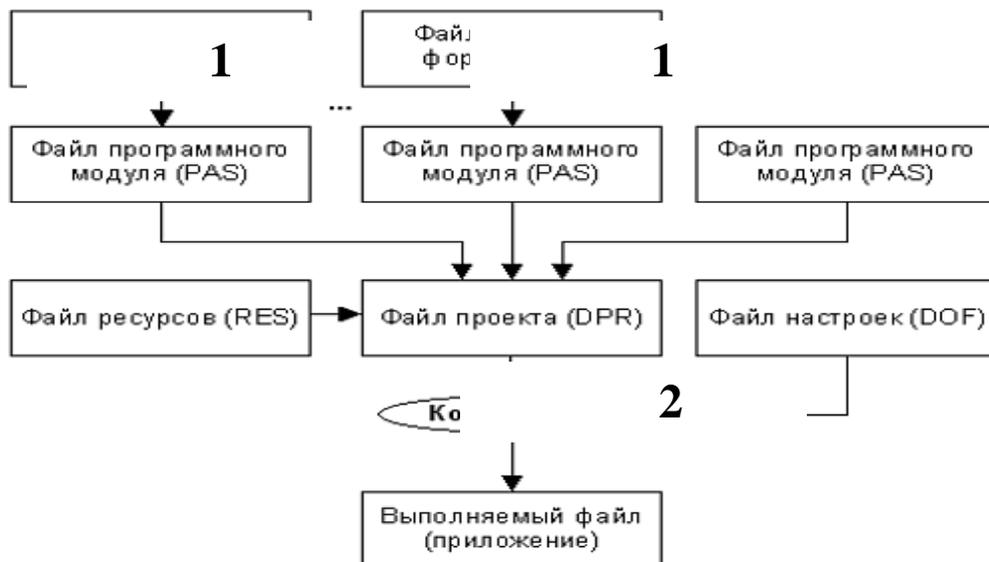
46. Модуль – это...

- 1) Средство для создания, изменения исходных файлов, которые содержат написанную на языке программирования программу
- 2) **Автономно компилируемая программная единица, включающая в себя различные компоненты раздела описаний (типы, константы, переменные, процедуры и функции) и, возможно некоторые исполняемые операторы иницилирующей части**
- 3) Набор инструментов, которые используются для преобразования символов

47. Установить соответствие

1) Файл проекта	a) (.pas) соответствующий файл модуль для хранения кода
2) Файл модуля	b) (.res) содержит пиктограмму и прочие ресурсы
3) Файл формы	c) (.dpr) текстовый файл используется для хранения информации о формах и модулях, содержит операторы инициализации и запуска программы на выполнение
4) Файл опций проекта	d) (.dfm) двоичный файл, который создается для хранения информации о ваших формах и фреймах
5) Файл ресурсов	e) (.cfg) хранит установки проекта
6) Файл конфигурации проекта	f) (.dof) хранит установки опций проекта
<b>1) – c, 2) – a, 3) – d, 4) – f, 5) – b, 6) – e</b>	

48. Заполнить схему:



- 1) 1 – файл формы (DFM); 2 – файл проекта (DPR)
- 2) 1 – файл проекта (DPR); 2 – файл формы (DFM)
- 3) 1 – файл формы (DPR); 2 – файл проекта (DFM)
- 4) 1 – файл проекта (DFM); 2 – файл формы (DPR)

49. Дополнить предложение. Набор всех файлов, необходимых для создания приложения, называется...
- 1) Модулем
  - 2) **Проектом**
  - 3) Задачей
  - 4) Формой
50. Файлы описания форм – это...
- 1) Отклики на внешнее воздействие
  - 2) Файлы подключающие все используемые программные модули
  - 3) **Текстовые файлы с расширением DFM, в котором сохраняются значения свойств формы и ее компонентов**
51. Содержимое dfm-файла вызывается:
- 1) **Контекстное меню формы -> View as Text**
  - 2) Контекстное редактора кода -> View as Form
  - 3) File -> Save as...
52. Дополнить предложение. При сборке приложения описание из DFM-файла помещается в...
- 1) Файл модуля
  - 2) **Область ресурсов**
  - 3) Область установок проекта
53. Программный модуль (Unit) содержит:
- 1) **Все относящиеся к форме объявления и методы обработки событий, написанные на языке Object Pascal**
  - 2) Заготовку, макет одного из окон разрабатываемого приложения
  - 3) Пиктограммы, которые представляют компоненты VCL, которые можно включить в приложение
54. Дополнить предложение. Имя модуля должно совпадать с...
- 1) Приложением Windows
  - 2) Спецификацией задачи
  - 3) Объектом в Delphi
  - 4) **Именем файла при сохранении File->Save as...**
55. Интерфейсная секция модуля (interface) содержит: (несколько вариантов ответа)
- 1) **Список стандартных модулей библиотеки VCL**

- 2) **Раздел описания типов**
  - 3) Компилятор
  - 4) **Раздел описания переменных (объявление собственно объекта формы)**
  - 5) Инспектор объектов и проектировщик форм
56. Дополнить предложение. Класс формы объявляется в разделе...
- 1) **Type**
  - 2) Uses
  - 3) Project
  - 4) Var
57. Дополнить предложение. В описании класса форм помещенные на форму компоненты представлены...
- 1) Границами формы
  - 2) **Полями формы**
  - 3) Индексами формы
58. Published – это...
- 1) **Атрибут видимости, позволяющий работать с компонентами и методами обработки событий на визуальном уровне**
  - 2) Директива препроцессора
  - 3) Текст программы на языке Object Pascal
59. Пустые секции, в которых можно размещать любые вспомогательные поля, методы и свойства: *(несколько вариантов ответа)*
- 1) **Private**
  - 2) Uses
  - 3) Var
  - 4) **Public**
  - 5) Program
60. Объявление объекта формы:
- 1) unit Unit1;
  - 2) type TForm1 = class(TForm)
  - 3) **var Form1: TForm1;**
61. implementation – это...
- 1) Тип данных Delphi
  - 2) Племя в Южной Африке
  - 3) **Раздел реализации объявлений интерфейса**
62. Подключение файла описания формы:
- 1) **{R \*.dfm}**
  - 2) {\$R \*.pas}
  - 3) {R \*.dpr}
63. Дополнить предложение. После подключения файла описания формы следует...
- 1) Раздел описания типов
  - 2) Атрибут видимости
  - 3) Описание размеров используемых компонентов
  - 4) **Реализация методов обработки событий**
64. Установить метод обработки события Button1Click: *(несколько вариантов ответа)*
- 1) **Двойной щелчок по компоненту Button1 на форме**
  - 2) **Выделить Button1 ->Events -> OnClick**
  - 3) Выделить BitBtn1 ->Events -> OnClick
65. Программный файл проекта – это...
- 1) Программист
  - 2) **Главная часть приложения, с которой начинается выполнение программы и которая обеспечивает инициализацию других модулей**
  - 3) Пустая секция

66. К комментариям не относятся: (несколько вариантов ответа)
- 1) **Program**
  - 2) {Form1 }
  - 3) **{ \$R \*.res }**
  - 4) //выполнение команды
67. Дополнить предложение. Сочетание end со следующей за ней точкой называется...
- 1) Терминалом программной единицы
  - 2) **Терминатором программной единицы**
  - 3) Андроидом программной единицы
68. Содержимое DPR-файла вызывается: (несколько вариантов ответа)
- 1) Ctrl+alt
  - 2) File -> New->Application
  - 3) **Project -> View Source**
  - 4) **Ctrl+F12**
69. Дополнить предложение. Зарезервированное слово **uses** сообщает компилятору о...
- 1) **Модулях, которые должны рассматриваться как необъемлемые части программы и которые содержатся в других файлах**
  - 2) Начале работы приложения
  - 3) Ошибках, которые могут возникнуть при компилировании
70. Модуль Forms – это...
- 1) Дополнительный модуль, содержащий неиспользуемые компоненты
  - 2) Модуль для построения более красочного приложения
  - 3) **Обязательный для всех программ модуль, который содержит определение объекта Application**
71. Запись `{ $R *.res }` означает:
- 1) Комментарий, указывающий программисту где искать ресурсы
  - 2) **Директиву, подключающую к результирующему выполняемому файлу ресурсы**
  - 3) Директиву, подключающую к результирующему выполняемому файлу форму
72. Установить соответствие:
- |                            |                                                        |
|----------------------------|--------------------------------------------------------|
| 1) Application.Initialize  | a) загружает и инициализирует форму                    |
| 2) Application.CreateForm  | b) активизирует форму и начинает выполнение приложения |
| 3) Application.Run         | c) подготавливает приложение к работе                  |
| <b>1 – c; 2 – a; 3 – b</b> |                                                        |
73. Включение множественного выбора элементов компонента ListBox1
- 1) ListBox2.MultiSelect:=false;
  - 2) **ListBox1.MultiSelect:=true;**
  - 3) ListBox1.MultiSelect:=false;
  - 4) ListBox1.Checked:=true;
74. Запись `ListBox1.Selected[3]:=true` означает что:
- 1) Необходимо установить три дополнительных элемента
  - 2) Необходимо удалить три лишних элемента
  - 3) **Выделен четвертый элемент в компоненте**
  - 4) Выделен третий элемент в компоненте
75. Простейшая функция вывода на экран текстовых сообщений:
- 1) ListBox1.Items.Add();
  - 2) **ShowMessage();**
  - 3) Case...of;
  - 4) CheckBox1.Checked:=true;
76. Полная очистка списка:

- 1) **ListBox1.Clear;**
  - 2) ListBox1.Items.Add();
  - 3) RadioGroup1.ItemIndex:=0;
77. Снятие «флажка» компонента CheckBox1
- 1) ListBox1.Checked:=true;
  - 2) **CheckBox1.Checked:=false;**
  - 3) CheckBox1.Checked:=true;
  - 4) CheckBox1.Caption :='Снять флажок';
78. Свойство Align=AllBottom:
- 1) Компонент перемещается в центр формы
  - 2) Компонент перемещается в верхнюю часть контейнера
  - 3) **Компонент перемещается в нижнюю часть контейнера и его ширина становится равной ширине контейнера**
79. Объект ComboBox - ...
- 1) Многострочный редактор
  - 2) **Комбинированный список**
  - 3) Функциональная кнопка
80. Установка в ComboBox1 первой строки массива Item:
- 1) **ItemIndex=0**
  - 2) ItemIndex=-1
  - 3) ItemIndex=1
  - 4) ItemIndex=356
81. Объект RadioGroup - ...
- 1) Команда «Авторадио»
  - 2) **Панель группы радиокнопок**
  - 3) Метка
82. Дополнить предложение. Компонент-флажок, который используется для обозначения включения или отключения какой либо опции, называется...
- 1) RadioGroup
  - 2) Функциональная кнопка
  - 3) **CheckBox**
  - 4) ListBox

### **МДК 01.03 Разработка мобильных приложений**

#### *Проверочные тесты*

1. Какие методы отражают жизненный цикл Activity:
  - onCreate() (+)
  - onRestart()(+)
  - onResume()(+)
  - onRunning()
  - onClose()
2. Подробная информация о приложении содержится в файле:
  - AndroidManifest.xml (+)
  - main.xml
  - R.java
  - default.properties
3. Для создания всплывающего уведомления необходимо инициализировать объект:
  - Toast (+)
  - Message
  - TextView
  - MessageBox
4. Единица измерения dp или dip - это 1/72 дюйма, определяется по физическому размеру экрана дюйм, определяется по физическому размеру экрана абстрактная ЕИ, поз-

воляющая приложениям выглядеть одинаково на различных экранах и разрешениях (+)  
физический элемент матрицы дисплея

5. Тип верстки при котором позиционирование элементов происходит относительно друг друга и относительно главного контейнера

AbsoluteLayout

FrameLayout

LinearLayout

RelativeLayout (+)

6. Какой класс можно использовать для перехода между Activity?

ActivityChanger

Activity Intent (+)

Switcher

7. От какого класса наследуются все элементы управления?

Control

Controls

Element

View (+)

8. Для чего используется класс R?

Класс ресурсов (+) Класс для управления элементами управления Класс для работы с потоками Класс, предназначенный только для доступа к идентификаторам элементов

9. Как программно получить доступ к элементам управления через идентификатор?

getViewById(id)

findViewById(id) (+)

findViewById(id)

getView(id)

10. В какой библиотеке находится класс Activity?

android

android.Activities

android.app (+)

java.app

11. На основе какой операционной системы была разработана система Android:

Linux (+)

MiniX

Windows

iOS

12. Как называется режим использования Android-системы с максимальными правами (аналог администратора Windows):

Суперпользователь (+)

Root-Home пользователь

Android user

Up-User

13. Как называется событие-клик объекта-кнопки:

OnClick (+)

OnClicked

Click

ClickEvent

14. Для добавления в приложение более одного экрана взаимодействия необходимо создание:

Activity (+)

View

Layout

XML-файла

15. Как называется класс для работы с SQLite-базами в Android:

SQLiteOpenHelper (+)

SQLProvider

SQLiteProvider

SQLiteContext

16. Как называется язык разметки Window Phone приложений:

XAML (+)

XML

Java

C#

17. С помощью какого класса можно вывести сообщение в Window Phone приложении:

Toast

MessageBox (+)

Message

ShowClass

18. Какой элемент предоставляет вариант разметки, элементы, помещенные в который, располагаются в стек горизонтально или вертикально:

StackPanel (+)

Pivot

Panorama

Canvas

19. Какой элемент позволяют позиционировать элементы внутри себя максимально гибко предоставляя возможность размещать элементы, используя строки и столбцы.

Grid (+)

Pivot

Panorama

Canvas

20. Какой элемент предоставляет наиболее простой вариант разметки. Он может быть использован для абсолютного позиционирования элементов с использованием координат.

Grid

Pivot

Panorama

Canvas (+)

21) На базе какого ядра сделано ядро Android?

1. Windows NT

2. Linux

3. FreeBSD

4. VxWorks

22) Какое название носит виртуальная Java-машина Android?

1. Dalvik

2. Bionic

3. HotSpot

23) Как изменился набор консольных утилит Linux в Android?

1. Существенно уменьшился

2. Остался без изменений

3. Существенно увеличился

4. В Android нет консольных утилит

24) Для чего предназначен Binder?

1. Связывает файлы ресурсов с приложением

2. Обеспечивает межпроцессное взаимодействие

3. Обмен информацией между приложениями
- 25) Является ли logcat полным аналогом gdb?
1. Да
  2. logcat — полный аналог gdbserver 3.Нет
- 26) Из-за чего glibc была заменена на Bionic?
1. Смена бренда
  2. Несовместимость glibc со свободными лицензиями
  3. glibc невозможно скомпилировать для ARM
  4. Уменьшение размеров объектного кода
- 27) Какой байт-код используется в Dalvik?
1. Обычный
  2. Исключительно свой собственный
  3. Свой собственный, но обычный может быть в него сконвертирован
- 28) Из чего состоит Java API в Android?
1. Часть стандартных библиотек Java
  2. Стандартные библиотеки Java и пакет javax.
  3. Набор стандартных, популярных и собственных пакетов
  4. Исключительно собственные библиотеки Android
- 29) Какую функциональность обеспечивает пакет android.view?
1. Реализация GUI
  2. Работа с камерой
  3. Темы рабочего стола
- 30) Какой способ беспроводной связи есть в пакете android.net?
1. Bluetooth
  2. WiFi
  3. NFC
  4. DRM

### **3. МАТЕРИАЛЫ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

#### **МДК 01.01 Разработка программных модулей**

Форма промежуточной аттестации: **дифференцированный зачет.**

#### *Вопросы к дифференцированному зачету*

1. Программы и программное обеспечение
2. Система счисления. Двоичные числа
3. Система счисления. Шестнадцатеричные числа
4. Алгоритм Банкира
5. Алгоритм Медника
6. Алгоритм замещения страниц
7. Исследование файловой системы ввода/вывода
8. Директивы определения данных
9. Директивы распределения памяти
10. Регистры
11. Архитектура микропроцессоров
12. Предложения языка Ассемблер
13. Операнды команд
14. Алгоритмы работы Ассемблеров
15. Описание данных в Ассемблер
16. Команды пересылки общего назначения
17. Команды загрузки адресных значений и обращения к стеку
18. Команды ввода/вывода
19. Арифметические операции над двоичными числами. Сложение (вычитание) двоичных чисел без знака

20. Арифметические операции над двоичными числами. Сложение (вычитание) двоичных чисел со знаком
  21. Арифметические операции над двоичными числами. Умножение двоичных чисел
  22. Арифметические операции над двоичными числами. Деление двоичных чисел
  23. Арифметические операции над десятичными числами. Сложение десятичных чисел
  24. Арифметические операции над десятичными числами. Умножение десятичных чисел
  25. Арифметические операции над десятичными числами. Деление десятичных чисел
  26. Логические команды. Команды логических операций.
  27. Логические команды. Команды сравнения.
  28. Логические команды. Команды сдвигов.
  29. Организация подпрограмм. Передача параметров в подпрограммы.
  30. Организация макрокоманд. Передача параметров в макрокоманды
- МДК 01.02. Прикладное программирование**
1. Назовите элементы интерфейса программы
  2. Сформулируйте технологию ввода кода программы
  3. Перечислите этапы алгоритма сохранения и запуска проекта
  4. Сформулируйте назначение вкладок «Свойства», «События»
  5. Назовите основные свойства компонентов «LABEL», «BUTTON»
  6. Назначение целочисленных типов данных
  7. Назначение вещественного типа данных
  8. Назначение денежного типа данных
  9. Назначение вариантного типа данных
  10. Назначение символьного типа данных
  11. Назначение интервального типа данных
  12. Назначение перечисляемого типа данных
  13. Основные стандартные математические функции
  14. Основные свойства компоненты «EDIT»
  15. Формат записи составного оператора
  16. Формат записи условного оператора
  17. Форма записи оператора варианта
  18. Свойства компоненты «TListBox»
  19. Назначение компоненты TComboBox.
  20. Режимы работы компоненты «Поле со списком».
  21. Назначение компоненты TCheckBox.
  22. Назначение стандартных панелей сообщений.
  23. Назначение компоненты TScrollBar и ее свойства
  24. Назначение компоненты TPanel и ее свойства

#### **МДК 01.02 Поддержка и тестирование программных модулей**

Форма промежуточной аттестации – **дифференцированный зачет.**

##### ***Вопросы к дифференцированному зачету***

1. Введение: тестирование - способ обеспечения качества программного продукта
2. Основные понятия тестирования
3. Подходы к обоснованию истинности формул и программ и их связь с тестированием. Вопросы организации тестирования. Фазы тестирования, основные проблемы тестирования и поставлена задача выбора конечного набора тестов.
4. Требования к идеальному критерию тестирования и классы частных критериев. Особенности применения структурных и функциональных критериев на базе конкретных примеров. Особенности применения методов стохастического тестирования и метод оцен-

ки скорости выявления ошибок. Мутационный критерий и на примере иллюстрируется техника работы с ним.

5. Оценка оттестированности проекта: метрики и методика интегральной оценки

6. Графовые модели проекта, метрики оценки оттестированности проекта, приводятся примеры плоской и иерархической моделей проекта.

7. Особенности модульного тестирования, подходы к тестированию на основе потока управления, потока данных, динамические и статические методы при структурном подходе. Взаимосвязь сборки модулей и методов интеграционного тестирования. Подходы монолитного, инкрементального, нисходящего и восходящего тестирования. Рассматриваются особенности интеграционного тестирования в процедурном программировании.

8. Интеграционное тестирование и его особенности для объектно-ориентированного программирования

9. Модель объектно-ориентированной программы, использующая понятие Р-путей и ММ-путей. Оценки сложности тестирования и методика тестирования объектно-ориентированной программы. Рассматривается пример интеграционного тестирования.

10. Разновидности тестирования: системное и регрессионное тестирование

11. Автоматизация тестирования

12. Особенности промышленного тестирования

13. Документирование и оценка промышленного тестирования

14. Описываются особенности документирования тестовых процедур для ручных и автоматизированных тестов, описаний тестовых наборов и тестовых отчетов. Рассматривается жизненный цикл дефекта. Обсуждаются метрики, используемые при тестировании.

15. Регрессионное тестирование: цели и задачи, условия применения, классификация тестов и методов отбора

16. Регрессионное тестирование: разновидности метода отбора тестов

17. Регрессионное тестирование: методики, не связанные с отбором тестов и методики порождения тестов

18. Регрессионное тестирование: алгоритм и программная система поддержки

19. Описание тестируемой системы и ее окружения. Планирование тестирования

20. Модульное тестирование на примере классов

21. Интеграционное тестирование

22. Системное тестирование

23. Ручное тестирование

24. Автоматизация тестирования с помощью скриптов

25. Автоматическая генерация тестов на основе формального описания

26. Описание ручного тестирования

27. Автоматизация тестирования с помощью скриптов

28. Описание автоматической генерации MSC тестов

### **МДК 01.03 Разработка мобильных приложений**

Формы промежуточной аттестации: **дифференцированный зачет.**

#### **Примеры практико-ориентированных заданий**

**Задание 1.** Скачать Android SDK + Eclipse (Eclipse ADT Bundle) <http://developer.android.com/sdk/index.html>, а также последнюю версию Java (JDK) <http://www.oracle.com/technetwork/java/index-jsp-138363.html> (или просто набрать в yandex "JDK" и пройти по ссылке на oracle.com). Установить всё это. Создать новый проект, зайти в Android SDK и скачать какую-нибудь версию Android SDK (например, lollipop - 5.0.1), выделив все инструменты для работы с ней; сконфигурировать эмулятор (желательно эмулировать своё собственное мобильное устройство, включая версию андроид на нём в качестве Target SDK). Изменить TextView с надписью "Hello world" на какую-нибудь другую надпись по желанию. Запустить проект на эмуляторе и убедиться, что всё работает.

**Задание 2.** Создать новый проект, написать программу, которая выводит в элемент TextView надпись, введённую пользователем в текстовом поле EditText после нажатия на

кнопку Button. Помимо этого в Activity должен быть TextView с ФИО студента и группой. Запустить на эмуляторе и убедиться, что всё работает.

**Задание 3.** Создать приложение, которое состоит из нескольких activities. Первое activity содержит элемент TextView с названием или номером activity, текстовое поле EditText для ввода какой-то информации, кнопку Button с названием "Next" или "Перейти на 2 activity/экран/окно" или просто "2". Помимо этого в 1 activity должен быть TextView с ФИО студента и группой. После нажатия на эту кнопку происходит переход на второе activity, где содержится TextView с названием или номером activity, TextView с надписью что-то вроде "В первом окне вы напечатали:" и под ним - ещё один TextView с содержимым EditText с первого activity, и, разумеется, кнопка "1" или "Вернуться на 1 экран" или "Вернуться к вводу текста", нажав на которую пользователь может перейти обратно к 1 activity. Запустить на эмуляторе и убедиться, что всё работает.

**Задание 4.** В новом проекте написать приложение, работающее с разными темами/стилями. Сначала создать свой стиль и применить его к какому-нибудь интерфейвному элементу, затем - свою тему, которая применяется ко всем интерфейсным элементам. Приложение при этом должно выглядеть нестандартно, запустить на эмуляторе и убедиться, что всё работает. При возникновении ошибок открыть лог (CatLog) внизу, найти первую красную надпись и породить свою тему от той, которая требуется в этом красном сообщении.

**Задание 5.** Создать пользовательский (свой) список. Например, получить доступ в приложении к контактам (Permissions-закладка в AndroidManifest.xml) и скопировать контакты телефона в свой список, который отобразить после запуска приложения. Или создать свой список в виде твиттера (картинка+текст), элементы которого просто статически задать в массиве (как и картинки).

**Задание 6.** Создать приложение, содержащее анимированные интерфейсные элементы (например, увеличивающиеся при клике на них кнопки, вращающиеся TextView и т.д.).

**Задание 7.** Создать приложение, отображающее после запуска карты Google или какие-нибудь другие карты.

**Задание 8.** Создать собственный виджет с настройками. Например, виджет, который открывает какой-то сайт, адрес которого можно поменять в настройках.

**Задание 9.** Создать приложение, использующее опциональное меню (меню настроек) и контекстное меню для какого-нибудь интерфейсного элемента. Естественно, выбор пунктов меню должен что-то менять в интерфейсных элементах или их отображении! Например, очистить поле ввода через контекстное меню, или отобразить невидимые интерфейсные элементы через установку галок в опциональном меню.

**Задание 10.** Создать приложение, отображающее после некоторых действий (нажатия на кнопку, например, или проверки корректности ввода текста в EditText) диалоговое окно, свидетельствующее об ошибке/информирующее/предупреждающее пользователя о чём-то.

**Задание 11.** Создать приложение, помещающее по нажатию на кнопку какое-то сообщение со звуком в панель уведомлений/статус-панель на эмуляторе.

**Задание 12.** Создать приложение, по нажатию кнопки в котором проигрывается какой-то звук.

**Задание 13.** Создать приложение, при запуске которого проигрывается какое-то видео.

**Задание 14.** Создать приложение, при запуске которого активируется фотокамера телефона, производится снимок, и этот снимок помещается в Image View интерфейса приложения.

**Задание 15.** Создать приложение, работающее с SharedPreferences и сохраняющее настройки, а также работающее с БД SQLite - заполняющее БД по нажатию кнопки 1 с помощью EditText, и выводящее все записи этой БД в какой-нибудь интерфейсный элемент

ниже с помощью кнопки 2 (в виде списка, datagrid или просто правильно настроенного Text View).

### ***Вопросы к дифференцированному зачету***

1. История возникновения мобильных операционных систем
2. Основные этапы становления рынка мобильных приложений
3. Современное состояние рынка мобильных приложений
4. Классификация видов мобильных приложений
5. Преимущества использования мобильных приложений в сравнении с веб-приложениями
6. Недостатки использования мобильных приложений в сравнении с десктопными приложениями
7. Инструментальные среды разработки мобильных приложений для операционной системы Apple iOS
8. Инструментальные среды разработки мобильных приложений для операционной системы Android
9. Инструментальные среды разработки мобильных приложений для операционной системы Windows Phone
10. Структура операционной системы iOS
11. Структура операционной системы Android
12. Структура приложения iOS
13. Структура приложения Android
14. Основные требования к интерфейсу приложений iOS
15. Основные требования к интерфейсу приложений Android
16. Android-манифест
17. Взаимодействие Android-приложения с сетью
18. Работа Android-приложения с локальной базой данных
19. Считывание информации Android-приложением с XML-файла
20. Вызов приложения из другого приложения в ОС Android
21. Проблемы безопасности мобильных операционных систем
22. Бизнес-модели распространения мобильных приложений
23. Стратегия размещения приложения на Google Play
24. Стратегия размещения приложения на AppStore
25. Сравнительная характеристика современных мобильных операционных систем
26. HTML5 и мобильные приложения
27. Проблемы совместимости мобильных приложений со старыми версиями операционных систем
28. Проектирование мобильных приложений с использованием C++
29. Технологии фреймворков в проектировании мобильных приложений
30. Проблемы масштабирования СУБД в мобильных приложениях
31. Основные технологии виртуализации в инструментальных средах при создании мобильных приложений
32. Проблемы обеспечения безопасности в платных мобильных приложениях
33. Перспективы развития рынка мобильных приложений в России

### **МДК 01.04 Системное программирование**

Форма промежуточной аттестации: **дифференцированный зачет.**

#### ***Примерные практико-ориентированные задания***

1. Использование инструментальной среды разработки и проектирования BP Win
2. Перевод чисел из одной системы счисления в другую.
3. Работа с программой отладчиком ASMTTool. Разработка и отладка программ
4. Создание программ типа .COM и .EXE. Компиляция программ
5. Разработка и отладка программ арифметических операций для работы с целыми числами

6. Разработка и отладка программы с процедурой и использование псевдооператоров определения данных
7. Разработка и отладка программ с использованием сегментации памяти в реальном режиме.
8. Использование средств разработки резидентных программ и модификации программных прерываний MS/DOS
9. Использование прерываний MS/DOS
10. Получение заданного изображения на экране монитора путем непосредственного программирования системы видеовывода
11. Исследование взаимодействия процессора и внешних устройств
12. Разработка программы взаимодействия с консолью
13. Разработка программы с использованием прерываний BIOS для работы с экраном и клавиатурой. Разработка программы с использованием прерываний MS-DOS для работы с экраном, клавиатурой и файлами
14. Разработка модульной программы
15. Использование средств программирования на ассемблере персональных ЭВМ класса IBM-PC/XT
16. Разработка программы со строками
17. Использование средств макроопределений в ассемблере
18. Разработка программы с применением макроопределений и макрокоманд
19. Управление изменениями, тестированием и документированием с использованием технологий Rational

### *Примерные билеты к дифференцированному зачету*

#### **Билет 1**

1. Поясните понятие системного программирования.
2. Написать и исследовать работу программы пересылки байта из ячейки памяти программ с адресом 71H в регистр R1 банка 1 с использованием регистра DPTR и в регистр R2 банка 3 с использованием регистра PC. (Assembler).
3. Проверить является ли строка палиндромом. (Палиндром - это выражение, которое читается одинаково слева направо и справа налево). (Borland C++ Builder).

#### **Билет 2**

1. Приведите принципы функционирования систем программирования.
2. Исследовать команды записи в стек и считывания из стека: PUSH P1; PUSH 01; PUSH PSW; POP PSW; POP 01; POP P1 (Assembler).
3. Преобразовать строку таким образом, чтобы в ее начале были записаны слова, содержащие только цифры, потом слова, содержащие только буквы, а затем слова, которые содержат и буквы и цифры. (Borland C++ Builder).

#### **Билет 3**

1. Поясните назначение и функции компилятора.
2. Написать и исследовать работу подпрограммы возведения в квадрат числа в аккумуляторе (Assembler).
3. Преобразовать строку таким образом, чтобы буквы каждого слова в ней были отсортированы по возрастанию. (Borland C++ Builder).

#### **Билет 4**

1. Дайте определение понятиям загрузчика, отладчика, приведите функции загрузчика.
2. Написать и исследовать программу опроса содержимого двух младших разрядов порта P1 и перехода в зависимости от их состояния к одной из четырех подпрограмм, начальные адреса которых находятся в ячейках 21H, 25H, 3AH и 2FH. (Assembler).
3. Преобразовать строку таким образом, чтобы цифры каждого слова в ней были отсортированы по убыванию. (Borland C++ Builder).

### **Билет 5**

1. Изобразите блок-схему разработки программ в архитектуре «клиент-сервер».
2. Разработать и исследовать программы сложения операндов (Assembler).
3. Произвести пословный перевод всех слов строки. Подстановочный словарь может содержать не более 10 слов, можно не учитывать изменение форм слова. (Borland C++ Builder).

### **Билет 6**

1. Поясните механизм подготовки и выполнения программ на языке Ассемблер.
2. Разработать и исследовать программу умножения операндов (Assembler).
3. Структурированный тип: Фирма  
Поля: название фирмы, Ф.И.О. директора, количество сотрудников, адрес, уставной капитал.

#### Функции:

Ввод количества фирм с клавиатуры

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Вычисление среднего уставного капитала

Поиск структуры с заданными Ф.И.О. директора. (Borland C++ Builder).

### **Билет 7**

1. Перечислите и охарактеризуйте режимы работы микропроцессора.
2. Разработать и исследовать программу деления операндов (Assembler).
3. Структурированный тип: абонент телефонной связи.  
Поля: Ф.И.О. абонента, номер телефона, тип соединения (индивидуальный, спаренный, коллективного пользования), адрес владельца

#### Функции:

Ввод количества абонентов с клавиатуры

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Поиск абонента с заданным номером

Подсчет количества абонентов с телефоном индивидуального пользования. (Borland C++ Builder).

### **Билет 8**

1. Перечислите типы операторов языка ассемблер.
2. Исследовать и модернизировать программу формирования временной задержки (Assembler).
3. Структурированный тип: Перечень комплектующих деталей компьютера  
Поля: тип детали (например, видеокарта, процессор и т. п), марка, производитель, цена, срок гарантийного обслуживания

#### Функции:

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Вычисление средней стоимости одного компьютера

Поиск комплектующей детали с минимальным гарантийным сроком обслуживания

Вывод перечня комплектующих заданного типа. (Borland C++ Builder).

### **Билет 9**

1. Как происходит использование моделей памяти и сегментации при создании программ.
2. Исследовать программу возведения в квадрат числа (Assembler).
3. Структурированный тип: Средство связи  
Поля: тип, радиус действия, цена одной минуты связи.

#### Функции:

Заполнение массива структур с клавиатуры

Печать массива структур на экране  
Вычисление средней стоимости одного часа связи для средств заданного типа  
Поиск средства связи с максимальным радиусом действия. (Borland C++ Builder).

#### **Билет 10**

1. Перечислите и охарактеризуйте типовые ошибки программирования.  
2. Исследовать программу перевода двоичного числа в двоично-десятичное число (Assembler).

3. Структурированный тип: товар

Поля: Название, фирма-производитель, входная цена одной единицы товара, количество, торговая надбавка в процентах

Функции:

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Поиск товара с минимальной ценой заданного производителя

Вычисление общего количества товаров. (Borland C++ Builder).

#### **Билет 11**

1. Запишите и поясните общий формат машинной команды.  
2. Разработать и исследовать подпрограмму вычисления синуса угла в диапазоне 0 - 80° с шагом 10°. (Assembler).

3. Структурированный тип: читатель

Поля: Ф.И.О., номер билета, количество книг на руках, дата перерегистрации(число, месяц, год)

Функции:

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Поиск читателя с наиболее поздней датой перерегистрации

Вычисление общего количества книг на руках. (Borland C++ Builder).

#### **Билет 12**

1. Приведите структуру и особенности работы обработчиков прерываний.

2. Разработать программу десятичной коррекции (Assembler)

3. Структурированный тип: книга

Поля: автор, название, год издания, число страниц

Функции:

Заполнение массива структур с клавиатуры

Печать массива структур на экране

Подсчет книг заданного автора

Поиск книги с наиболее поздним годом издания. (Borland C++ Builder).

#### **Билет 13**

1. Как происходит организация взаимодействия резидентных программ.

2. Разработать программу перевода двоичного числа в двоично-десятичное, используя рекуррентное соотношение

$d7 \ d6 \ d5 \ d4 \ d3 \ d2 \ d1 \ d0 = d0 + 2(d1 + 2(d2 + 2(d3 + 2(d4 + 2(d5 + 2(d6 + 2 \ d7))))))$  (Assembler).

3. Структурированный тип: тарифный план сотовой связи

Поля: Название фирмы-провайдера, название тарифа, абонентская плата, количество бесплатных минут, входящих в абонентскую плату, стоимость минуты разговора в дневное и вечернее время

Функции:

Заполнение массива структур с клавиатуры. Печать массива структур на экране

Поиск тарифа с максимальной абонентской платой. Подсчет оптимального по цене тарифа с условием, что абонент планирует наговаривать в месяц не более некоторого числа минут, заданного с клавиатуры. (Borland C++ Builder).

#### **Билет 14**

1. Приведите порядок операций по защите программ от копирования и несанкционированного доступа.
2. Разработать программу перевода двоично-десятичного числа в двоичное (Assembler).
3. Написать программу просмотра и редактирования простейшей базы данных о студентах. Базу данных считывать из файла. Приложение должно состоять из двух диалогов. В первом диалоге редактируется число записей в базе, имя и фамилия одного (текущего) студента. Второй диалог предназначен для просмотра всей базы в виде таблицы. Высота таблицы должна автоматически вычисляться по текущему разрешению экрана. (Borland C++ Builder).

#### **Билет 15**

1. Как осуществляются арифметические операции над данными в ASCII - и BCD – форматах.
2. Разработать программу временной задержки для датчика движения (Assembler).
3. Написать программу простейшего графического редактора, позволяющего рисовать (добавлять, удалять, редактировать свойства) фигуры из разработанной системы классов в окне созданного приложения. Изучить структуру приложения Windows, использующего OpenGL. Изучить архитектуру, синтаксис команд, примитивы OpenGL. Написать программу рисования простых трехмерных тел. (Borland C++ Builder).

#### **Билет 16**

1. Приведите примеры мнемонической записи команд языка Ассамблер.
2. Разработать программу вычисления времени до завершения процесса (Assembler).
3. Описать класс точки с характеристиками (x, y) – вещественные координаты точки и методами ввода, вывода и нахождения расстояния от точки до начала координат. (Borland C++ Builder).

#### **Билет 17**

1. Приведите порядок операций отладки программных модулей с использованием специализированных программных средств.
2. Разработать программу управления семисегментным индикатором (Assembler).
3. Описать класс точки, добавив к нему методы возврата координат X и Y. С помощью этого класса решить следующие прикладные задачи: даны точки A(x1, y1) и B(x2, y2). Определить, лежат ли длина отрезков OA и OB (O – начало координат) в заданном интервале [m1, m2]. (Borland C++ Builder).

#### **Билет 18**

1. Поясните механизм взаимодействия Turbo Assembler и Borland Pascal.
2. Разработать программу определения координат в пространстве (Assembler).
3. Описать класс прямоугольника с характеристиками a и b – длины сторон, и методами нахождения периметра, площади и диагонали прямоугольника. Решить следующие задачи с использованием описанного класса: имеется прямоугольный участок со сторонами A и B. Определить стоимость работ по установке забора вокруг участка, если установки забора длиной 10 метров требует расходов в N рублей. (Borland C++ Builder).

#### **Билет 19**

1. Поясните основные правила синтаксиса языка C++.
2. Разработать программу определения расстояния до объекта (Assembler).
3. Описать класс параболы  $at^2+bt+c$  с характеристиками a, b и c, и методами определения координаты  $t_0$  – вершины параболы и значения функции  $y = at^2+bt+c$  в произвольной точке t. Для определения значения в вершине параболы воспользоваться тем, что значение производной в этой точке равняется нулю, то есть  $2at+b=0$ . Решить следующие задачи: спрос на продукцию предприятия снижается по левой ветви параболы  $t^2-4t+9$ . Определить, в какой момент времени  $t_0$  спрос будет минимальным и чему он будет равен (в тыс. ед.) (Borland C++ Builder).

### **Билет 20**

1. Приведите общие правила построения программ для Windows (на примере C++).
2. Разработать программу управления автомобильным тахометром (Assembler).
3. Имеются данные об успеваемости 10-ти студентов (средние баллы). Найти средний балл по группе. (Borland C++ Builder).

### **Билет 21**

1. Приведите пример простейшего приложения, реализующего обработку сообщений.
2. Имеется команда:  
ECS: 100 B8 04 30 05 00 30 CB  
Что делает данная программа?  
(Assembler).
3. Формируются несколько групп переключателей по 2 – 3 переключателя и статическое окно. В каждый момент времени только один из переключателей в группе может быть выбран. Информация об изменении состояния каждого переключателя должна отображаться в статическом окне. (Borland C++ Builder).

### **Билет 22**

1. Опишите функции обработки сообщений от драйвера «мыши», для C++ Builder.
2. Напишите программу, выполняющую следующую операцию:
  - Пересылка значения 25H в регистре AL
  - Сдвиг содержимого регистра AL на 1 бит влево
  - Пересылка значения 15H в регистр BL
  - Умножение содержимого регистра AL на содержимое регистра BL(Assembler).
3. Формируются несколько радиокнопок и статическое окно. В каждый момент времени только одна из радиокнопок должна быть нажата. Информация об изменении состояния каждой кнопки должна отображаться в статическом окне. (Borland C++ Builder).

### **Билет 23**

1. Приведите механизм отладки программных модулей с использованием специализированных программных средств.
2. Для сегментов кода, данных и стека даны имена CDSEA, DATSEA и STKSEA соответственно. Сформируйте директиву ASSUME. (Assembler).
3. Формируются однострочный текстовый редактор, простой список выбора и стандартная кнопка. Нажатие на кнопку должно помещать введенную пользователем в редакторе строку в список выбора. (Borland C++ Builder).

### **Экзаменационный билет № 24**

1. Приведите порядок операций для оптимизации программных модулей.
2. Напишите программу для вычисления 12 чисел Фибоначчи: 1,1,2,3,5,8,13,...(каждое число в последовательности представляет собой сумму двух предыдущих чисел). Для организации цикла используйте команду LOOP. (Assembler).
3. Формируются список выбора с окном редактирования и статическое окно. В статическом окне должна отображаться информация о выбираемом пользователем элементе списка. (Borland C++ Builder).

### **Билет 25**

1. Приведите порядок операций для тестирования программных модулей.
2. Напишите программу для вычисления суммы первых 15 нечетных чисел Фибоначчи с первыми 5 четными числами Фибоначчи. Суммы должны вычисляться в подпрограммах. (Assembler).
3. Формируются три стандартные кнопки и статическое окно. Две кнопки управляют размерами третьей: нажатие на них пропорционально увеличивает и уменьшает ее размеры. Информация об изменении размеров кнопки должна отображаться в статическом окне. (Borland C++ Builder).

### **Билет 26**

1. Таймеры в ОС Windows, связанные с ними сообщения, функции API Windows обработки этих событий.

2. Напишите программу вывода на экран набора символов ASCII – кода (Assembler).

3. Формируются несколько стандартных кнопок и статическое окно. Информация о нажатии каждой кнопки должна отображаться в статическом окне. (Borland C++ Builder).

### **Билет 27**

1. Опишите функции обработки клавиатурных сообщений в ОС Windows, функции API Windows, для C++.

2. Напишите программу, запрашивающую ввод имени, а затем отображающую в середине экрана введенное имя (Assembler).

3. Имеются данные об объемах продаж магазина за 12 месяцев. Найти суммарный и среднемесячный объемы продаж. (Borland C++ Builder).

### **Билет 28**

1. Поясните механизм взаимодействия Turbo Assembler и Borland C++.

2. Напишите команды для BIOS INT 13H, выполняющие чтение одного сектора в область памяти INDISK, с дисковод А, головки 0, дорожки 6 и сектора 3 (Assembler).

3. Имеются данные о численности 10 населенных пунктов района. Найти количество населенных пунктов, количество жителей в которых ниже среднего по району. (Borland C++ Builder).

### **Билет 29**

1. Приведите пример программирования операций над файлами, каталогами и дисками.

2. Напишите команды для BIOS INT 13H, выполняющие запись трех секторов из области памяти OUTDISK на дисковод В, головку 0, дорожку 8 и сектор 1 (Assembler).

3. Имеются данные о прибыли предприятия за последние пять лет. Найти, в какие годы прибыль была максимальной и минимальной. (Borland C++ Builder).

### **Билет 30**

1. Расшифруйте понятия загрузчики и редакторы связей.

2. Программа MAINPRO должна вызвать подпрограмму SUBPRO. В программе MAINPRO определены переменные QTY как DB, VALUE как DW и PRICE как DW. Подпрограмма SUBPRO должна разделить VALUE на QTY и записать частное а PRICE. Постройте работающую программу и проверьте ее (Assembler).

3. Имеются данные о температуре воздуха за неделю (по дням), определить перепад температур за неделю. (Borland C++ Builder).

## **ПМ.04 Разработка модулей программного обеспечения для компьютерных систем**

### **Форма промежуточной аттестации – экзамен по модулю**

Экзамен в форме собеседования по практическому заданию:

- 1) по построению алгоритма в соответствии с техническим заданием
- 2) по разработке программного модуля в соответствии с техническим заданием
- 3) по выполнению отладки предложенного программного модуля
- 4) выполнению заданных видов тестирования программного модуля
- 5) по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга
- 6) по созданию модуля для заданного мобильного устройства на основе спецификации

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОФЕССИОНАЛЬНОГО МОДУЛЯ

Код и наименование профессиональных и общих компетенций, формируемых в рамках модуля	Критерии оценки	Методы оценки
<b>МДК 01.01 Разработка программных модулей</b>		
<p>ПК 1.1 Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием</p>	<p>Оценка «отлично» - техническое задание проанализировано, алгоритм разработан, соответствует техническому заданию и оформлен в соответствии со стандартами, пояснены его основные структуры, указаны использованные стандарты в области документирования; выполнена оценка сложности алгоритма</p> <p>Оценка «хорошо» - алгоритм разработан, оформлен в соответствии со стандартами и соответствует заданию, пояснены его основные структуры, выполнена оценка сложности алгоритма</p> <p>Оценка «удовлетворительно» - алгоритм разработан и соответствует заданию.</p>	<p>Экзамен/дифференцированный зачет/зачет в форме собеседования: практическое задание по построению алгоритма в соответствии с техническим заданием</p> <p>Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> <p>Экзамен по модулю</p>

<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «отлично» - программный модуль (для квалификации "Специалист по тестированию в области информационных технологий": тестовый модуль) разработан по имеющемуся алгоритму в среде разработки, на указанном языке программирования) методами объектно-ориентированного/структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.</p> <p>Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Экзамен/дифференцированный зачет/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием</p> <p>Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> <p>Экзамен по модулю</p>
<p><b>МДК 01.02 Поддержка и тестирования программных модулей</b></p>		

<p>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Оценка «зачтено»</p> <ul style="list-style-type: none"> <li>- выполнена отладка модуля с использованием инструментария среды проектирования); с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки.</li> <li>- выполнена отладка модуля с использованием инструментария среды проектирования); сохранены и представлены результаты отладки.</li> <li>- выполнена отладка модуля, пояснены ее результаты.</li> </ul>	<p>Зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля</p> <p>Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> <p>Экзамен по модулю</p>
<p>ПК 1.4 Выполнять тестирование программных модулей</p>	<p>Оценка «зачтено»</p> <ul style="list-style-type: none"> <li>- выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования в соответствии со стандартами.</li> <li>- выполнено тестирование модуля, в том числе с помощью инструментальных средств, и оформлены результаты тестирования</li> <li>- выполнено тестирование модуля и оформлены результаты тестирования.</li> </ul>	<p>Зачет в форме собеседования: практическое задание по выполнению заданных видов тестирования программного модуля.</p> <p>Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> <p>Экзамен по модулю</p>

<p>ПК 1.5 Осуществлять рефакторинг и оптимизацию программного кода</p>	<p>Оценка «зачтено»</p> <ul style="list-style-type: none"> <li>- определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на уровнях переменных, функций, классов, алгоритмических структур; проведена оптимизация и подтверждено повышение качества программного кода.</li> <li>- определены качественные характеристики программного кода с помощью инструментальных средств; выявлены фрагменты некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</li> <li>- определены качественные характеристики программного кода частично с помощью инструментальных средств; выявлено несколько фрагментов некачественного кода; выполнен рефакторинг на нескольких уровнях; проведена оптимизация и выполнена оценка качества полученного программного кода.</li> </ul>	<p>Зачет в форме собеседования: практическое задание по оценке качества кода предложенного программного модуля, поиску некачественного программного кода, его анализу, оптимизации методами рефакторинга.</p> <p>Защита отчетов по лабораторным работам</p> <p>Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики</p> <p>Экзамен по модулю</p>
<p><b>МДК 01.03 Разработка мобильных приложений</b></p>		

<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «зачтено»:  Оценка «отлично» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.  Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.  Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно-ориентированного/структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов</p>	<p>Дифференцированный зачет/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием  Защита отчетов по лабораторным работам  Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики  Экзамен по модулю</p>
--------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>ПК 1.6 Разрабатывать модули программного обеспечения для мобильных платформ.</p>	<p>Оценка «зачтено»:  Оценка «отлично» - разработан модуль для заданного мобильного устройства с соблюдением основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено его соответствие спецификации.  Оценка «хорошо» - разработан модуль для заданного мобильного устройства с учетом основных этапов разработки на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие выполняемых функций спецификации с незначительными отклонениями.  Оценка «удовлетворительно» - разработан модуль для заданного мобильного устройства на одном из современных языков программирования; при проверке работоспособности модуля на устройстве или эмуляторе установлено соответствие основных выполняемых функций спецификации.</p>	<p>Дифференцированный зачет/зачет в форме собеседования: практическое задание по созданию модуля для заданного мобильного устройства на основе спецификации  Защита отчетов по лабораторным работам  Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики  Экзамен по модулю</p>
<p><b>МДК 01.04 Системное администрирование</b></p>		

<p>ПК 1.2 Разрабатывать программные модули в соответствии с техническим заданием</p>	<p>Оценка «зачтено»:  Оценка «отлично» - программный модуль) разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и полностью соответствует техническому заданию, соблюдены и пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам.  Оценка «хорошо» - программный модуль разработан по имеющемуся алгоритму в среде разработки на указанном языке программирования методами объектно- ориентированного/ структурного программирования и практически соответствует техническому заданию с незначительными отклонениями, пояснены основные этапы разработки; документация на модуль оформлена и соответствует стандартам  Оценка «удовлетворительно» - программный модуль разработан по имеющемуся алгоритму в среде разработки методами объектно- ориентированного/ структурного программирования и соответствует техническому заданию; документация на модуль оформлена без существенных отклонений от стандартов.</p>	<p>Дифференцированный зачет/зачет в форме собеседования: практическое задание по разработке программного модуля в соответствии с техническим заданием  Защита отчетов по лабораторным работам  Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики  Экзамен по модулю</p>
--------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>ПК 1.3 Выполнять отладку программных модулей с использованием специализированных программных средств</p>	<p>Оценка «зачтено»: Оценка «отлично» - выполнена отладка модуля с использованием инструментария среды проектирования; с пояснением особенностей отладочных классов; сохранены и представлены результаты отладки. Оценка «хорошо» - выполнена отладка модуля с использованием инструментария среды проектирования; сохранены и представлены результаты отладки. Оценка «удовлетворительно» - выполнена отладка модуля, пояснены ее результаты.</p>	<p>Дифференцированный зачет/зачет в форме собеседования: практическое задание по выполнению отладки предложенного программного модуля Защита отчетов лабораторным работам Интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики Экзамен по модулю</p>
<p>ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.</p>	<p>обоснованность постановки цели, выбора и применения методов и способов решения профессиональных задач; - адекватная оценка и самооценка эффективности и качества выполнения профессиональных задач</p>	<p>Экспертное наблюдение за выполнением работ</p>
<p>ОП 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.</p>	<p>использование различных источников, включая электронные ресурсы, медиаресурсы, Интернет-ресурсы, периодические издания по специальности для решения профессиональных задач</p>	
<p>ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.</p>	<p>демонстрация ответственности за принятые решения обоснованность самоанализа и коррекция результатов собственной работы;</p>	
<p>ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.</p>	<p>взаимодействовать с обучающимися, преподавателями и мастерами в ходе обучения, с руководителями учебной и производственной практик; обоснованность анализа работы членов команды (подчиненных)</p>	

<p>ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста</p>	<p>Демонстрировать грамотность устной и письменной речи, - ясность формулирования и изложения мыслей</p>	
<p>ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей.</p>	<p>соблюдение норм поведения во время учебных занятий и прохождения учебной и производственной практик,</p>	
<p>ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях</p>	<p>эффективное выполнение правил ТБ во время учебных занятий, при прохождении учебной и производственной практик; - демонстрация знаний и использование ресурсосберегающих технологий в профессиональной деятельности</p>	
<p>ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.</p>	<p>эффективность использовать средств физической культуры для сохранения и укрепления здоровья при выполнении профессиональной деятельности.</p>	
<p>ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.</p>	<p>эффективность использования информационно-коммуникационных технологий в профессиональной деятельности согласно формируемым умениям и получаемому практическому опыту; - эффективность использования в профессиональной деятельности необходимой технической документации, в том числе на английском языке.</p>	