

**ГАОУ ВО «ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ НАРОДНОГО ХОЗЯЙСТВА»**

*Утверждены решением  
Ученого совета,  
протокол № 10  
от 30 мая 2017 г.*

**КАФЕДРА «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И  
ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»**

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ**

**ПО ДИСЦИПЛИНЕ  
«ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»**

**НАПРАВЛЕНИЕ ПОДГОТОВКИ – 09.03.03 ПРИКЛАДНАЯ  
ИНФОРМАТИКА, ПРОФИЛЬ «ПРИКЛАДНАЯ  
ИНФОРМАТИКА В ЭКОНОМИКЕ»**

**Уровень высшего образования - бакалавриат**

**УДК 004.6 (075.32)**

**ББК 32.973я723**

**Составитель** – Кобзаренко Дмитрий Николаевич, профессор кафедры «Информационные технологии и информационная безопасность» ДГУНХ.

**Внутренний рецензент** – Савина Елена Владимировна, кандидат физико-математических наук, доцент кафедры «Информационные технологии и информационная безопасность».

**Внешний рецензент** – Абдурагимов Гусейн Эльдарханович, кандидат физико-математических наук, доцент кафедры "Математические методы в экономике" Дагестанского государственного университета.

**Представитель работодателя** - Представитель работодателя - Сайидахмедов Сайидахмед Сергеевич, генеральный директор компании «Текама».

*Оценочные материалы по дисциплине «Технологии и методы программирования» разработаны в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика», утвержденного приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г., № 207, в соответствии с приказом от 5 апреля 2017г., № 301 Министерства образования и науки РФ.*

Оценочные материалы по дисциплине «Технологии и методы программирования» размещены на официальном сайте [www.dgunh.ru](http://www.dgunh.ru)

Кобзаренко Д.Н. Оценочные материалы по дисциплине «Технологии и методы программирования» для направления подготовки 09.03.03 Прикладная информатика, профиль «Прикладная информатика в экономике». – Махачкала: ДГУНХ, 2017 г. – 31 с.

Рекомендованы к утверждению Учебно-методическим советом ДГУНХ 29 мая 2017 г.

Рекомендованы к утверждению руководителем основной профессиональной образовательной программы высшего образования – программы бакалавриата по направлению подготовки 09.03.03 Прикладная информатика, профиль «Прикладная информатика в экономике», к.э.н., доцент Раджабов К.Я.

Одобрены на заседании кафедры «Информационные технологии и информационная безопасность» 25 мая 2017 г., протокол № 10.

## Содержание

НАЗНАЧЕНИЕ ОЦЕНОЧНЫХ МАТЕРИАЛОВ .....	4
РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины.....	5
1.1 Перечень формируемых компетенций.....	5
1.2 Перечень компетенций с указанием видов оценочных средств .....	5
РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине.....	9
РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания .....	24
РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций .....	28
Лист актуализации оценочных материалов по дисциплине .....	31

## НАЗНАЧЕНИЕ ОЦЕНОЧНЫХ МАТЕРИАЛОВ

Оценочные материалы для текущего контроля успеваемости (оценивания хода освоения дисциплин), для проведения промежуточной аттестации (оценивания промежуточных и окончательных результатов обучения по дисциплине) обучающихся по дисциплине «Технологии и методы программирования» на соответствие их учебных достижений поэтапным требованиям образовательной программы высшего образования 09.03.03 Прикладная информатика, профиль «Прикладная информатика в экономике».

Оценочные материалы по дисциплине «Технологии и методы программирования» включают в себя: перечень компетенций с указанием этапов их формирования в процессе освоения ОП ВО; описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания; типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения ОПОП; методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Оценочные материалы сформированы на основе ключевых принципов оценивания:

- валидности: объекты оценки должны соответствовать поставленным целям обучения;
- надежности: использование единообразных стандартов и критериев для оценивания достижений;
- объективности: разные обучающиеся должны иметь равные возможности для достижения успеха.

Основными параметрами и свойствами оценочных материалов являются:

- предметная направленность (соответствие предмету изучения конкретной дисциплины);
- содержание (состав и взаимосвязь структурных единиц, образующих содержание теоретической и практической составляющих дисциплины);
- объем (количественный состав оценочных материалов);
- качество оценочных материалов в целом, обеспечивающее получение объективных и достоверных результатов при проведении контроля с различными целями.

## РАЗДЕЛ 1. Перечень компетенций с указанием видов оценочных средств в процессе освоения дисциплины

### 1.1 Перечень формируемых компетенций

код компетенции	формулировка компетенции
<b>ПК</b>	<b>ПРОФЕССИОНАЛЬНЫЕ КОМПЕТЕНЦИИ</b>
<b>ПК-2</b>	способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение
<b>ПК-12</b>	способностью проводить тестирование компонентов программного обеспечения ИС

### 1.2 Перечень компетенций с указанием видов оценочных средств

<i>Формируемые компетенции</i>	<i>Планируемые результаты обучения по дисциплине, характеризующие этапы формирования компетенций</i>	<i>Уровни освоения компетенции</i>	<i>Критерии оценивания сформированности компетенций</i>	<i>Виды оценочных средств</i>
<b>ПК-2:</b> способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение	<u><b>Знать:</b></u> Особенности структурного и объектно-ориентированного программирования. Методы организации структур данных, сортировки и поиска.	Пороговый уровень	Обучающийся слабо (частично) знает особенности структурного и объектно-ориентированного программирования, методы организации структур данных, сортировки и поиска.	<b>Блок А</b> – задания репродуктивного уровня - тестовые задания; - вопросы для обсуждения.
		Базовый уровень	Обучающийся с незначительными ошибками и отдельными пробелами знает особенности структурного и объектно-ориентированного программирования, методы организации структур данных, сортировки и поиска.	
		Продвинутый уровень	Обучающийся с требуемой степе-	

			<p>нью полноты и точности знает особенности структурного и объектно-ориентированного программирования, методы организации структур данных, сортировки и поиска.</p>	
<p><b><u>Уметь:</u></b> Выполнять декомпозицию задачи на подпрограммы, модули, объекты. Применять методы организации структур данных, сортировки, поиска при создании программ.</p>	Пороговый уровень	Обучающийся слабо (частично) умеет выполнять декомпозицию задачи на подпрограммы, модули, объекты, применять методы организации структур данных, сортировки, поиска при создании программ.	<p><b>Блок В</b> – задания для реконструктивного уровня - задачи; - лабораторные задания.</p>	
	Базовый уровень	Обучающийся с незначительными затруднениями умеет выполнять декомпозицию задачи на подпрограммы, модули, объекты, применять методы организации структур данных, сортировки, поиска при создании программ.		
	Продвинутый уровень	Обучающийся умеет выполнять декомпозицию задачи на подпрограммы, модули, объекты, применять методы организации структур данных, сортировки, поиска при создании программ.		
<p><b><u>Владеть:</u></b> Навыками структурного и объектно-ориентированного программирования. Навыками при-</p>	Пороговый уровень	Обучающийся слабо (частично) владеет навыками структурного и объектно-ориентированного программирования, навыками приме-	<p><b>Блок С</b> – задания для практико-ориентированного уровня - индивидуальные проекты.</p>	

	менения методов организации структур данных, сортировки, поиска.		нения методов организации структур данных, сортировки, поиска.	
		Базовый уровень	Обучающийся с незначительными затруднениями владеет навыками структурного и объектно-ориентированного программирования, навыками применения методов организации структур данных, сортировки, поиска.	
		Продвинутый уровень	Обучающийся свободно владеет навыками структурного и объектно-ориентированного программирования, навыками применения методов организации структур данных, сортировки, поиска.	
<b>ПК-12:</b> способностью проводить тестирование компонентов программного обеспечения ИС	<b><u>Знать:</u></b> Как тестировать разработанные подпрограммы и классы.	Пороговый уровень	Обучающийся слабо (частично) знает как тестировать разработанные подпрограммы и классы.	<b>Блок А</b> – задания репродуктивного уровня - тестовые задания; - вопросы для обсуждения.
		Базовый уровень	Обучающийся с незначительными ошибками и отдельными пробелами знает как тестировать разработанные подпрограммы и классы.	
		Продвинутый уровень	Обучающийся с требуемой степенью полноты и точности знает как тестировать разработанные подпрограммы и классы.	
	<b><u>Уметь:</u></b> Тестировать	Пороговый уровень	Обучающийся слабо (частично) умеет	<b>Блок В</b> – задания для реконструк-

разработанные подпрограммы и классы.		тестировать разработанные подпрограммы и классы.	тивного уровня - задачи; - лабораторные задания.
	Базовый уровень	Обучающийся с незначительными затруднениями умеет тестировать разработанные подпрограммы и классы.	
	Продвинутый уровень	Обучающийся умеет тестировать разработанные подпрограммы и классы.	
<b>Владеть:</b> Навыками тестирования разработанных подпрограмм и классов.	Пороговый уровень	Обучающийся слабо (частично) владеет навыками тестирования разработанных подпрограмм и классов.	<b>Блок С</b> – задания для практико-ориентированного уровня - индивидуальные проекты.
	Базовый уровень	Обучающийся с незначительными затруднениями владеет навыками тестирования разработанных подпрограмм и классов.	
	Продвинутый уровень	Обучающийся свободно владеет навыками тестирования разработанных подпрограмм и классов.	



## РАЗДЕЛ 2. Задания, необходимые для оценки планируемых результатов обучения по дисциплине

Для проверки сформированности компетенции ПК-2: способность разрабатывать, внедрять и адаптировать прикладное программное обеспечение

### Блок А. Задания репродуктивного уровня («знать»)

#### А.1 Фонд тестовых заданий по дисциплине

**1. Какой главный недостаток имеет архитектура программы с подпрограммами и глобальной областью данных?**

- А. Вероятность замедления скорости выполнения программы.
- Б. Вероятность нехватки оперативной памяти при отладке программы.
- В. Вероятность переполнения стека при использовании большого числа глобальных переменных.
- Г. Вероятность искажения части переменных в глобальных данных какой-либо подпрограммой.**

**2. Какой подход был предложен для того, чтобы сократить количество ошибок в архитектуре программы с глобальной областью данных и подпрограммами?**

- А. Объектно-ориентированный подход.
- Б. Подпрограммы с локальной областью данных.**
- В. Увеличение размера памяти для глобальных переменных.
- Г. Дублирование глобальных переменных.

**3. Что является основой структурного программирования?**

- А. Разбиение сложных систем на части с целью последующей реализации в виде небольших подпрограмм.**
- Б. Разработка структур данных для программы.
- В. Выделение основных структурных элементов программы: ветвления и циклы.
- Г. Структурный язык программирования.

**4. Модульное программирование заключается в:**

- А. Произвольном разбиении программного кода на отдельные файлы-модули.
- Б. Декомпозиции решаемой задачи на отдельные мелкие части – модули.
- В. Выделении групп подпрограмм, использующих одни и те же глобальные данные в отдельно компилируемые модули.**
- Г. Том, что в программе выделяются главный модуль и подмодули.

**5. При проектировании программного обеспечения «сверху-вниз» вначале будут спроектированы:**

- А. Элементы интерфейса пользователя.

Б. Локальные подпрограммы.

**В. Модули.**

Г. Объекты.

**6. При проектировании программного обеспечения «снизу-вверх» вначале будут спроектированы:**

А. Элементы интерфейса пользователя.

**Б. Локальные подпрограммы.**

В. Модули.

Г. Объекты.

**7. Если мы хотим, чтобы глобальная переменная F модуля MyUnit была видна из других программных модулей, то:**

А. она объявляется в исполнительной части модуля.

Б. она объявляется со специальной директивой видимости.

В. она объявляется динамической.

**Г. она объявляется в интерфейсной части модуля.**

**8. Если мы хотим, чтобы глобальная переменная F модуля MyUnit не была видна из других программных модулей, то:**

**А. она объявляется в исполнительной части модуля.**

Б. она объявляется со специальной директивой видимости.

В. она объявляется динамической.

Г. она объявляется в интерфейсной части модуля.

**9. Пусть имеется объект «Студент» в рамках блочно-иерархического подхода. Какой из перечисленных блоков усиливает конкретизацию объекта (находится на более низком уровне, чем остальные)?**

А. Фамилия.

Б. Год рождения.

В. Пол.

**Г. Средняя оценка за 1 курс.**

**10. Пусть имеется объект «Преподаватель» в рамках блочно-иерархического подхода. Какой из перечисленных блоков усиливает абстракцию (находится на более высоком уровне, чем остальные)?**

**А. Фамилия.**

Б. Нагрузка по дисциплине «технологии и методы программирования»

В. Нагрузка на заочном факультете.

Г. Среднегодовая зарплата за 2018г.

**11. Модель жизненного цикла программного обеспечения в рамках которой переход на следующую стадию выполняется после завершения предыдущего этапа называется:**

**А. Спиральной.**

Б. С промежуточным контролем.

**В. Каскадной.**

Г. Последовательной.

**12. Выберите ложные утверждения:**

А. Связное распределение памяти для хранения линейного списка требует дополнительного пространства памяти, по сравнению с последовательным распределением.

**Б. При последовательном распределении памяти для хранения линейного списка легче включить/исключить элемент по сравнению со связным распределением.**

В. При связном распределении памяти для хранения линейного списка медленнее, чем при последовательном, выполняются обращения к произвольным частям списка.

Г. При использовании связного распределения памяти для хранения линейного списка по сравнению с последовательным усложняется задача объединения двух списков или разбиения списка на части.

**13. Методика разработки программ, в основе которой лежит понятие объекта как некоторой структуры, описывающей объект реального мира, его поведение – это ...**

**А. Объектно-ориентированное программирование**

Б. Объект

В. Инкапсуляция

Г. Наследование

**14. Некоторая часть окружающего нас мира, которая может быть рассмотрена как единое целое, - это ...**

А. Объектно-ориентированное программирование

**Б. Объект**

В. Инкапсуляция

Г. Полиморфизм

**15. Возможность скрыть внутреннее устройство объекта от его пользователей, предоставив через интерфейс доступ только к тем членам объекта, с которыми клиенту разрешается работать напрямую – это ...**

А. Объектно-ориентированное программирование

Б. Объект

**В. Инкапсуляция**

Г. Полиморфизм

**16. Возможность при описании класса указывать на его происхождение от другого класса - это ...**

А. Производная

Б. Инкапсуляция

## **В. Наследование**

Г. Полиморфизм

**17. Возможность объектов с одинаковой спецификацией иметь различную реализацию – это...**

А. Двойственность

Б. Инкапсуляция

В. Наследование

**Г. Полиморфизм**

**18. Именованные категории, позволяющие группировать сходные объекты – это ...**

**А. Классы**

Б. Объекты

В. События

Г. Свойства

**19. Отдельные, четко обозначенные экземпляры некоторого класса - это ...**

А. Классы

**Б. Объекты**

В. События

Г. Свойства

**20. Некоторые особые состояния, в которые может попадать объект – это ...**

А. Объекты

**Б. События**

В. Свойства

Г. Подпрограммы

**21. Характеристики объекта – это ...**

А. Объекты

Б. События

**В. Свойства**

Г. Методы класса

**22. Функции, объявление которых включено в описание класса, выполняющие действия над объектами класса – это ...**

А. Объекты

Б. События

В. Свойства

Г. Методы класса

## **А.2 Перечень вопросов**

1. Каковы основные этапы развития технологии программирования?

2. В чем проблемы разработки сложных программных систем?
3. В чем состоит блочно-иерархический подход к созданию сложных программных систем?
4. Что представляет собой жизненный цикл и этапы разработки ПО?
5. Что представляет собой эволюция моделей жизненного цикла ПО?
6. Как оценить качество процессов создания ПО?
7. Что такое понятие технологичности ПО?
8. Что такое модули и какие у них свойства?
9. Чем отличаются нисходящая и восходящая разработка ПО?
10. Чем отличаются структурное и неструктурное программирование?
11. Что такое линейные информационные структуры?
12. В чем заключается последовательное распределение памяти при хранении линейных списков?
13. В чем заключается связанное распределение памяти при хранении линейных списков?
14. Что представляют собой ортогональные списки?
15. Что такое деревья и как их представлять?
16. Как проходить деревья?
17. Как представлять леса бинарными деревьями?
18. Как использовать понятия дерева для решения прикладных задач?
19. Как оценить сложность алгоритмов?
20. Что такое сортировка?
21. Как выполняется внутренняя сортировка?
22. Как выполняется сортировка подсчетом?
23. Как выполняется сортировка вставками?
24. Как выполняется обменная сортировка?
25. Как выполняется сортировка посредством выбора?
26. Что такое последовательный поиск?
27. Как выполнять поиск в упорядоченной таблице?
28. Как выполнять поиск по бинарному дереву?
29. Что такое сбалансированные деревья?
30. Что такое Хеширование?
31. Перечислите базовые конструкции структурного программирования в современных языках?
32. Что представляют собой записи, списки, массивы, указатели, функции.
33. Что представляют собой классы и объекты.
34. Что такое конструктор класса?
35. Что такое деструктор класса?
36. Для чего в классах нужны указатели this или self?
37. Как происходит наследование классов?

## **Блок В. Задания реконструктивного уровня («уметь»)**

### **В.1 Фонд типовых задач по дисциплине**

1. Даны три целочисленные переменные A, B, C. Кроме того, имеется СТЕК (хранящий максимум 3 целочисленных значения) и две функции работы с ним: PUSH(k) – заносит значение, передаваемое через параметр k в стек и POP – извлекает значение из стека и возвращает его как результат выполнения функции. Напишите алгоритм, который реализует сортировку данных в переменных A, B, C в порядке возрастания и не использует никаких других переменных и структур данных. Также запрещается присваивать значения между переменными A, B, C.

2. Условие предыдущей задачи только СТЕК меняется на ОЧЕРЕДЬ.

3. Нарисуйте дерево у которого имеется 4 узла (A, B, C, D), корень находится в вершине A. Узлы имеют следующие значения степени:

A = 2, B = 0, C = 1, D = 0.

4. Нарисуйте бинарное дерево, соответствующее следующей записи ( B ( C ) ( A ( E ( F ) ) ( D ( G ) ) ) ) ).

5. Представьте следующее арифметическое выражение в виде бинарного дерева:  
 $\frac{2x}{y} (9 + \sqrt{5 - d})$ .

6. Определите сложность следующего алгоритма на языке C++ (в зависимости от n):

```
int s = 0;
for (int i = 1, i <= n, i++) {
    int j = 1;
    do {
        s = s + i * j;
        j++;
    } while j <= n;
}
```

7. Реализуйте приведенный ниже программный код на Delphi виде класса:

```
program primer;
var N, i, s: Integer;
    A: array of Integer;
begin
    ReadLn(N);
    SetLength(A, N);
    for i:= 0 to N-1 do ReadLn(A[i]);
    s:= 0;
    for i:= 0 to N-1 do
        s:= s + A[i];
    WriteLn(s);
end.
```

## **В.2 Лабораторные задания**

### **Лабораторная работа №1. Линейные списки.**

**Цель работы:** получить практические навыки программирования с линейными структурами данных

**Задачи:**

1. Изучить теоретические вопросы построения структур данных на основе линейных списков.
2. Написать программу, которая предоставляет возможность работы со списком элементов (записей) с двумя вариантами: последовательным и связным распределением памяти. Программа выполняет операции по добавлению элемента в список, удалению элемента из списка и редактированию элемента в списке.

### **Лабораторная работа №2. Алгоритмы сортировки.**

**Цель работы:** познакомиться с известными алгоритмами сортировки и понять принципы их работы.

**Задачи:**

1. Изучить теоретические вопросы связанные с алгоритмами сортировки.
2. Написать программу, которая выполняет сортировку массива несколькими способами (сортировка подсчетом, сортировка вставками, обменная сортировка, сортировка посредством выбора), а также тестирует алгоритмы на большом наборе данных и выбирает лучший алгоритм по времени выполнения.

### **Лабораторная работа №3. Алгоритмы поиска.**

**Цель работы:** познакомиться с алгоритмами поиска и получить практические навыки применения алгоритма поиска на основе бинарного поиска.

**Задачи:**

1. Изучить теоретические вопросы, связанные с алгоритмами поиска и бинарным алгоритмом поиска.
2. Написать программу, которая для списка выполняет сортировку его элементов и затем выполняет поиск элемента по ключевому значению. Ключи поиска могут меняться.

### **Лабораторная работа №4. Объектно-ориентированное программирование**

**Цель работы:** получить практические навыки по созданию классов и объектов в современных языках программирования.

**Задачи:**

1. Изучить основы объектно-ориентированного программирования в языках программирования C++, Delphi, C#, Java, Python.
2. Реализовать алгоритмы из лабораторных работ №2 и 3 в виде класса на, как минимум 2 языках программирования.

## **Блок С. Задания практикоориентированного уровня для диагностирования сформированности компетенций («владеть»)**

### **С.1 Индивидуальные проекты**

1. Написать программу, решающую поставленную задачу (**типовые темы индивидуальных проектов приводятся в конце раздела**), в которой описаны входные и выходные данные. Решение задачи оформлять **ИСКЛЮЧИТЕЛЬНО** в виде класса.
2. Для **ТЕСТИРОВАНИЯ** класса создается объект-экземпляр класса.
3. Использовать **ЛЮБОЙ** широко известный язык программирования, который позволяет писать программы с помощью объектно-ориентированного подхода.

## **Блок Д. Задания для использования в рамках промежуточной аттестации**

### **Д.1 Перечень вопросов на экзамен**

1. Проблемы разработки и блочно-иерархический подход к созданию сложных программных систем.
2. Жизненный цикл и этапы разработки ПО. Эволюция моделей жизненного цикла ПО.
3. Понятие технологичности ПО. Модули и их свойства. Нисходящая и восходящая разработка ПО.
4. Структурное и неструктурное программирование. Программирование с защитой от ошибок. Сквозной структурный контроль.
5. Линейные информационные структуры. Последовательное распределение памяти при хранении линейных списков.
6. Связанное распределение памяти при хранении линейных списков. Массивы и ортогональные списки.
7. Деревья и их представление. Прохождение деревьев. Представление лесов бинарными деревьями.
8. Понятие сортировки. Внутренняя сортировка. Сортировка подсчетом. Сортировка вставками.
9. Обменная сортировка. Сортировка посредством выбора. Сортировка слиянием. Распределяющая сортировка. Внешняя сортировка.
10. Последовательный поиск. Поиск в упорядоченной таблице. Поиск по бинарному дереву.
11. Базовые конструкции структурного программирования на языках C++ / Delphi (на выбор).
12. Основные принципы ООП. Понятие классов и объектов в языках C++/C# / Java / Delphi / Python (на выбор).
13. Абстрактные классы и методы, перегрузка методов. Конструктор и деструктор.
14. Понятие интерфейса, стека, кучи и сборщика мусора в языке Java.



Для проверки сформированности компетенции ПК-12: способностью проводить тестирование компонентов программного обеспечения ИС

**Блок А. Задания репродуктивного уровня («знать»)**

**А.1 Фонд тестовых заданий по дисциплине**

**1. Какого типа ошибок в программе не существует?**

- А. синтаксические;
- Б. грамматические;**
- В. ошибки времени выполнения;
- Г. алгоритмические.

**2. Какие ошибки называют ошибками времени компиляции?**

- А. синтаксические;**
- Б. грамматические;
- В. ошибки времени выполнения;
- Г. алгоритмические.

**3. Какие ошибки называют исключениями?**

- А. синтаксические;
- Б. грамматические;
- В. ошибки времени выполнения;**
- Г. алгоритмические.

**4. Какие ошибки являются наиболее легко устранимые?**

- А. синтаксические;
- Б. грамматические;
- В. ошибки времени выполнения;
- Г. алгоритмические.**

**5. При каких ошибках программа не компилируется?**

- А. синтаксических;**
- Б. грамматических;
- В. ошибок времени выполнения;
- Г. алгоритмических.

**6. Часть программы, которая начинает свое выполнение в случае возникновения определенной исключительной ситуации....**

- А. прерывание;
- Б. вектор прерываний;
- В. обработчик исключений;**
- Г. подпрограмма.

**7. Для чего используются следующие конструкции в Delphi « try {защищаемый код} finally {код завершения} end;» и «try {защищаемый код} except {обработчик исключений} end;»?**

- А. для обработки текущих событий;
- Б. для диагностики критических ситуаций;
- В. для нормального завершения программы;
- Г. для обработки исключительных ситуаций в программе.**

**8. Какие ошибки обнаруживает компилятор:**

- А. синтаксические;**
- Б. грамматические;
- В. ошибки времени выполнения;
- Г. алгоритмические.

**9. Ключевое слово, обозначающее, что далее следуют инструкции, при выполнении которых возможно возникновение исключений, и что обработку этих исключений берет на себя программа:**

- А. Except;
- Б. On;
- В. Try;**
- Г. Else.

**10. Ключевое слово в Delphi, за которым следует тип исключения:**

- А. Except;
- Б. On;**
- В. Try;
- Г. Else.

**11. Ключевое слово в Delphi, обозначающее начало секции обработки исключений:**

- А. Except;**
- Б. On;
- В. Try;
- Г. Else.

**12. Опасный участок кода программы в Delphi размещается между словами:**

- А. On и Except;
- Б. Try и Else;
- В. Try и Except;**
- Г. On и Else.

**13. Класс исключений в Delphi, который генерируется при ошибке преобразования строк или объектов:**

- А. EConvertError;**
- Б. Exception;
- В. EComponentError;

Г. InvalidOperationException.

**14. Какое ключевое слово в языке Java и C# используется при выбросе исключения?**

A. raise

**Б. throw**

B. error

Г. exception

**15. Какое ключевое слово в языках Python и Delphi используется при выбросе исключения?**

**A. raise**

Б. throw

B. error

Г. exception

## **A.2 Перечень вопросов**

1. Что подразумевает программирование с защитой от ошибок?
2. Что представляет собой сквозной структурный контроль?
3. Что представляет собой механизм обработки исключений?
4. Как реализован механизм обработки исключений в языке C++?
5. Как реализован механизм обработки исключений в языке Delphi?
6. Как реализован механизм обработки исключений в языках Java и C#?
7. Как реализован механизм обработки исключений в языке Python?
8. Какие виды ошибок существуют в программном коде?
9. Как выполняется отладка программы при компиляции?
10. Как правильно выполнить тестирование разработанного приложения?

## **Блок В. Задания реконструктивного уровня («уметь»)**

### **В.1 Фонд типовых задач по дисциплине**

1. Пусть имеется программа, в которой пользователь должен ввести 6 чисел:  $a_1$ ,  $b_1$ ,  $c_1$ ,  $a_2$ ,  $b_2$ ,  $c_2$ , которые являются исходными данными для решения системы двух линейных уравнений:  $a_1 \cdot x + b_1 \cdot y = c_1$  и  $a_2 \cdot x + b_2 \cdot y = c_2$ . Перечислите три исключительные ситуации, которые могут возникнуть в работе программы.

2. Пусть имеется программный код, который решает некоторую задачу. Например, *«Вводится целое  $N$ , затем вводится последовательность из  $N$  целых чисел. Требуется найти и вывести все двузначные числа, сумма квадратов цифр которых делится на 17»*. Напишите тестовые примеры входных данных, которые позволят выявить все возможные ошибки в программе.

3. Дана программа на Delphi найдите явную ошибку в программном коде (вычисление факториала), которая приведет к исключительной ситуации и невозможности выполнить задачу.

```
program Factorial_Example;
uses SysUtils, Windows;
type
  TFactorial = class
  private
    FN: Integer;
  procedure SetN(Value: Integer);
  public
    property N: Integer read FN write SetN;
    function Calculate: Integer;
  end;
  procedure TFactorial.SetN(Value: Integer);
  begin
    if Value < 0 then FN:= 0
    else
      FN:= Value;
    end;
  function TFactorial.Calculate: Integer;
  var i: integer;
  begin
    Result:= 1;
    for i:= 1 to FN do Result:= Result * i;
  end;
var Factorial: TFactorial;
    k: Integer;
begin
  WriteLn('Введите целое число для вычисления факториала:');
  ReadLn(k);
  Factorial.N:= k;
  k:= Factorial.Calculate;
  WriteLn('Результат = ', k);
  Factorial.Free;
  ReadLn;
end.
```

## **В.2 Лабораторные задания**

### **Лабораторная работа №5. Обработка исключительных ситуаций.**

**Цель работы:** Приобретение навыков использования операторов, связанных с исключительными ситуациями в программах.

**Задачи:**

- 1) Создать программу, которая будет вычислять корни квадратного уравнения  $ax^2+bx+c=0$ .
- 2) Использовать операторы для обработки исключений, возникающих в программе (проверка корректности ввода коэффициентов  $a$ ,  $b$ ,  $c$  и существования корней уравнения).
- 3) Протестировать и отладить приложение.

### **Блок С. Задания практикоориентированного уровня для диагностирования сформированности компетенций («владеть»)**

#### **С.1 Индивидуальные проекты**

Для индивидуальных проектов (перечень тем приводится в конце раздела) разработать тестовые примеры, которые позволяют смоделировать все возможные ситуации в работе программы с целью выявить все возможные ошибки при тестировании.

### **Блок Д. Задания для использования в рамках промежуточной аттестации**

#### **Д.1 Перечень вопросов на экзамен**

1. Исключительные ситуации и механизмы их обработки в современных языках программирования.
2. Виды и причины ошибок в программе. Отладка программы и создание тестовых наборов для проверки корректности работы.

### **ПЕРЕЧЕНЬ ТЕМ ИНДИВИДУАЛЬНЫХ ПРОЕКТОВ**

#### **Требования к выполнению индивидуального проекта:**

- Написать программу, решающую поставленную задачу, в которой описаны входные и выходные данные. Решение задачи оформлять **ИСКЛЮЧИТЕЛЬНО** в виде класса.
- Для **ТЕСТИРОВАНИЯ** класса создается объект-экземпляр класса.
- Использовать **ЛЮБОЙ** язык программирования, который позволяет писать программы с помощью объектно-ориентированного подхода.
- Прислать исходный код преподавателю только в случае успешного решения.
- Распечатать исходный код на бумаге и ответить на вопросы, как и что было сделано.

#### **Темы:**

1. Дано целое положительное число  $n$ . Вывести  $n$  чисел Фибоначчи.

2. Даны целые положительные числа  $a$ ,  $b$ ,  $n$ . Вывести  $n$  членов арифметической прогрессии начиная с  $a$  и разностью прогрессии  $b$ .
3. Даны целые положительные числа  $a$ ,  $b$ ,  $n$ . Вывести  $n$  членов геометрической прогрессии начиная с  $a$  и знаменателем прогрессии  $b$ .
4. Дано целое положительное число  $n$ . Вывести  $n$  простых чисел, начиная с 2.
5. Даны целые положительные числа  $d$  и  $m$ . Первое соответствует числу, а второе – номеру месяца календарного года. Вывести каким по счету днем является день указанной даты. Например, 2 февраля это 33-й день.
6. Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на  $p$  процентов от имеющейся суммы ( $p$  – вещественное число,  $2 < p < 20$ ). По данному  $p$  определить, через сколько месяцев размер вклада превысит 1100 руб., и вывести найденное количество месяцев  $k$  (целое число) и итоговый размер вклада  $s$  (вещественное число сумму).
7. Дан целочисленный массив целых чисел  $n$ . Проверить, образуют ли его элементы арифметическую прогрессию. Если да, то вывести разность прогрессии, если нет – вывести 0.
8. Дан целочисленный массив целых чисел  $n$ . Проверить, образуют ли его элементы геометрическую прогрессию. Если да, то вывести знаменатель прогрессии, если нет – вывести 0.
9. Дан целочисленный массив целых чисел  $n$ . Проверить, образуют ли его элементы возрастающую последовательность. Если да, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
10. Дан массив целых чисел размера  $n$ . Проверить, чередуются ли в нем четные и нечетные числа (0 считать четным числом). Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
11. Дан массив целых чисел размера  $n$ . Проверить, чередуются ли в нем положительные и отрицательные числа (0 считать положительным числом). Если чередуются, то вывести 0, если нет, то вывести номер первого элемента, нарушающего закономерность.
12. Дан целочисленный массив размера  $n$ . Вывести все его локальные минимумы и их количество, затем локальные максимумы и их количество.
13. Дан целочисленный массив размера  $n$ . Определить количество промежутков монотонного возрастания (то есть участков, на которых его элементы возрастают).

14. Дан целочисленный массив размера  $n$ . Определить сколько этот массив содержит простых чисел и вывести простые числа.
15. Дана строка. Подсчитать количество содержащихся в ней цифр, прописных букв и строчных букв.
16. Дана строка. Преобразовать все строчные буквы в прописные.
17. Дана строка. Преобразовать все прописные буквы в строчные.
18. Дана строка-предложение на русском языке. Преобразовать строку так, чтобы каждое слово начиналось с заглавной буквы.
19. Дана строка, состоящая из русских слов, разделенных пробелами (одним или несколькими). Вывести строку, содержащую эти же слова (разделенные одним пробелом), но расположенные в обратном порядке.
20. Дана строка-предложение на русском языке. Вывести самое длинное слово в предложении (если таких слов несколько, то вывести первое из них).
21. Дана строка-предложение, содержащая избыточные пробелы. Преобразовать ее так, чтобы между словами был ровно один пробел.
22. Дана строка выражение. Определить равно ли количество открывающихся скобок количеству закрывающихся и выдать сообщение типа: «Количества открывающихся и закрывающихся скобок равны», «Количество открывающихся скобок на 1 превышает количество закрывающихся» и т.д.
23. Дано целое число в диапазоне 100 – 999. Вывести строку – словесное описание данного числа, например: 256 – «двести пятьдесят шесть», 814 – «восемьсот четырнадцать».
24. Дано целое число в диапазоне 20 – 69, определяющее возраст (в годах). Вывести строку – словесное описание указанного возраста, обеспечив правильное согласование числа со словом «год», например: 20 – «двадцать лет», 32 – «тридцать два года», 41 – «сорок один год».
25. Дано целое число  $X$  в диапазоне 0-999, которое соответствует сумме в рублях. В зависимости от заданного числа записать ее значение прописью. Например:  $X=0$  (нуль рублей),  $X=109$  (сто девять рублей),  $X=931$  (девятьсот тридцать один рубль).
26. Дано целое число, лежащее в диапазоне от –999 до 999. Вывести строку – словесное описание данного числа вида "отрицательное двузначное число", "нулевое число", "положительное однозначное число" и т.д.

### **РАЗДЕЛ 3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

Балльно-рейтинговая система является базовой системой оценивания сформированности компетенции обучающихся очной формы обучения.

Итоговая оценка сформированности компетенции обучающихся в рамках балльно-рейтинговой системы осуществляется в ходе текущего контроля успеваемости, промежуточной аттестации и определяется как сумма баллов, полученных обучающимися в результате прохождения всех форм контроля.

Оценка сформированности компетенции по дисциплине складывается из двух составляющих:

✓ первая составляющая – оценка преподавателем сформированности компетенции в течение семестра в ходе текущего контроля успеваемости (максимум 100 баллов). Структура первой составляющей определяется технологической картой дисциплины, которая в начале семестра доводится до сведения обучающихся;

✓ вторая составляющая – оценка сформированности компетенции обучающихся на экзамене (максимум – 30 баллов).

Для студентов очно-заочной и заочной форм обучения применяются 4-балльная шкала оценивания результатов текущего контроля успеваемости и промежуточной аттестации обучающихся.

<b>уровни освоения компетенций</b>	<b>продвинутый уровень</b>	<b>базовый уровень</b>	<b>пороговый уровень</b>	<b>допороговый уровень</b>
<b>100 – балльная шкала</b>	85 и $\geq$	70 – 84	51 – 69	0 – 50
<b>4 – балльная шкала</b>	«отлично»	«хорошо»	«удовлетворительно»	«неудовлетворительно»

#### **Шкала оценок при текущем контроле успеваемости по различным показателям**

<b>Показатели оценивания сформированности компетенций</b>	<b>Баллы</b>	<b>Оценка</b>
<b>выполнение тестов</b>	0-30	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»
<b>ответы на устные вопросы</b>	0-15	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»
<b>выполнение лабораторных заданий</b>	0-15	«неудовлетворительно»



		«удовлетворительно» «хорошо» «отлично»
<b>выполнение индивидуальных проектов</b>	0-10	«неудовлетворительно» «удовлетворительно» «хорошо» «отлично»

**Соответствие критериев оценивания уровню освоения компетенций по текущему контролю успеваемости**

<i><b>Баллы</b></i>	<i><b>Оценка</b></i>	<i><b>Уровень освоения компетенций</b></i>	<i><b>Критерии оценивания</b></i>
0-50	«неудовлетворительно»	Допороговый уровень	Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины
51-69	«удовлетворительно»	Пороговый уровень	Не менее 50% заданий, подлежащих текущему контролю успеваемости, выполнены без существенных ошибок
70-84	«хорошо»	Базовый уровень	Обучающимся выполнено не менее 75% заданий, подлежащих текущему контролю успеваемости, или при выполнении всех заданий допущены незначительные ошибки; обучающийся показал владение навыками систематизации материала и применения его при решении практических заданий; задания выполнены без ошибок
85-100	«отлично»	Продвинутый уровень	100% заданий, подлежащих текущему контролю успеваемости, выполнены самостоятельно и в требуемом объеме; обучающийся проявляет умение обобщать, систематизировать материал и применять его при решении практических заданий; задания выполнены с подробными пояснениями и аргументированными выводами

### Шкала оценок по промежуточной аттестации

<i>Наименование формы промежуточной аттестации</i>	<i>Баллы</i>	<i>Оценка</i>
Экзамен	25-30	«отлично»
	19-24	«хорошо»
	12-18	«удовлетворительно»
	0-11	«неудовлетворительно»

### Соответствие критериев оценивания уровню освоения компетенций по промежуточной аттестации обучающихся

<i>Баллы</i>	<i>Оценка</i>	<i>Уровень освоения компетенций</i>	<i>Критерии оценивания</i>
0-11	«неудовлетворительно»	Допороговый уровень	Обучающийся не приобрел знания, умения и не владеет компетенциями в объеме, закрепленном рабочей программой дисциплины; обучающийся не смог ответить на вопросы
12-18	«удовлетворительно»	Пороговый уровень	Обучающийся дал неполные ответы на вопросы, с недостаточной аргументацией, практические задания выполнены не полностью, компетенции, осваиваемые в процессе изучения дисциплины сформированы не в полном объеме.
19-24	«хорошо»	Базовый уровень	Обучающийся в целом приобрел знания и умения в рамках осваиваемых в процессе обучения по дисциплине компетенций; обучающийся ответил на все вопросы, точно дал определения и понятия, но затрудняется подтвердить теоретические положения практически-ми примерами; обучающийся показал хорошие знания по предмету, владение навыками систематизации материала и полностью выполнил практические задания
25-30	«отлично»	Продвинутый уровень	Обучающийся приобрел знания, умения и навыки в полном объеме, закрепленном рабочей программой дисциплины; терминологический аппарат использован правильно; ответы полные,

			обстоятельные, аргументированные, подтверждены конкретными примерами; обучающийся проявляет умение обобщать, систематизировать материал и выполняет практические задания с подробными пояснениями и аргументированными выводами
--	--	--	---

#### РАЗДЕЛ 4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков, характеризующие этапы формирования компетенций

В вопросы к экзамену включены два теоретических вопроса и практическое задание, соответствующие содержанию формируемых компетенций. Экзамен проводится в письменной форме. На ответ и решение задачи студенту отводится 60 минут. За ответ на теоретические вопросы студент может получить максимально – 20 баллов, за решение задачи максимально – 10 баллов.

Тестирование имеет статус контрольной работы, выполняется в письменном виде на тестовых бланках. На выполнение тестирования дается 90 минут.

##### Методика оценивания выполнения тестов

Баллы	Оценка	Показатели	Критерии
25-30	«отлично»	1. Полнота выполнения тестовых заданий; 2. Своевременность выполнения;	Выполнено 85 % заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос.
19-24	«хорошо»	3. Правильность ответов на вопросы; 4. Самостоятельность тестирования.	Выполнено 70 % заданий предложенного теста, в заданиях открытого типа дан полный, развернутый ответ на поставленный вопрос; однако были допущены неточности в определении понятий, терминов и др.
12-18	«удовлетворительно»		Выполнено 50 % заданий предложенного теста, в заданиях открытого типа дан неполный ответ на поставленный вопрос, в ответе не присутствуют доказательные примеры, текст со стилистическими и орфографическими ошибками.
0-11	«неудовлетворительно»		Выполнено до 50% заданий предложенного теста, на поставленные вопросы ответ отсутствует или неполный, допущены существенные ошибки в теоретическом материале (терминах, понятиях).

Устный опрос проводится на практических занятиях. Студенту дается вопрос на основе пройденного материала из перечня вопросов на зачет. Студент должен выйти к доске и дать развернутый ответ на поставленный вопрос.

##### Методика оценивания ответов на устные вопросы

Баллы	Оценка	Показатели	Критерии
14-15	«отлично»	1. Полнота данных ответов; 2. Аргументиро-	Полно и аргументировано даны ответы по содержанию задания. Обнаружено понимание материала, может обосновать свои

		ванность данных ответов; 3. Правильность ответов на вопросы.	суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные. Изложение материала последовательно и правильно.
10-13	«хорошо»		Студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «отлично», но допускает 1-2 ошибки, которые сам же исправляет.
6-9	«удовлетворительно»		Студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки.
0-5	«неудовлетворительно»		Студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.

Темы индивидуальных проектов предлагаются на выбор в начале семестра. Студент выполняет проект и по окончании его выполнения высылает на проверку преподавателю. После проверки соответствия проекта и содержимого теме, а также правильности выполнения назначается день защиты. Защита проекта происходит в рамках регламента: 10-15 минут выступление, далее ответы на вопросы по теме проекта.

### Методика оценивания выполнения индивидуальных проектов

Баллы	Оценка	Показатели	Критерии
9-10	«отлично»	1. Полнота выполнения проекта; 2. Своевременность выполнения проекта;	Выполнены все требования к выполнению проекта; разработана, отлажена и протестирована программа; даны правильные ответы на дополнительные вопросы.
6-8	«хорошо»	3. Правильность ответов на вопросы	Выполнены основные требования к проекту, имеются недочеты в разработке и тестировании программы; на дополнительные вопросы при защите даны неполные ответы.

3-5	«удовлетворительно»	сы; 4. Самостоятельность выполнения проекта.	Требования к проекту выполнены не полностью, программа разработана, но в ней имеются существенные недостатки; допущены фактические ошибки при ответе на дополнительные вопросы.
0-2	«неудовлетворительно»		Проект не выполнен; обнаруживается существенное непонимание в том, как его выполнять.

Лабораторные задания выполняются на лабораторных занятиях. Задания выполняются на основе методических указаний к лабораторным работам. Студент должен выполнить лабораторное задание во временном интервале 60 минут. Остальные 30 минут даются на защиту лабораторного задания в рамках которой проверяется правильность и самостоятельность его выполнения.

### Методика оценивания выполнения лабораторных заданий

Баллы	Оценка	Показатели	Критерии
14-15	«отлично»	1. Полнота выполнения лабораторного задания; 2. Своевременность выполнения задания;	Задание выполнено самостоятельно. При этом составлен правильный алгоритм решения, в логических рассуждениях и в решении нет ошибок, получен верный ответ, задание решено рациональным способом.
10-13	«хорошо»	3. Последовательность и рациональность выполнения задания; 4. Самостоятельность решения.	Задание выполнено с помощью преподавателя. При этом составлен правильный алгоритм решения, в логических рассуждениях и решении нет существенных ошибок; есть объяснение решения, но задание решено нерациональным способом или допущено не более двух несущественных ошибок, получен верный ответ.
6-9	«удовлетворительно»		Задание выполнено с подсказками преподавателя. При этом задание понято правильно, в логическом рассуждении нет существенных ошибок, но допущены существенные ошибки в алгоритмах и/или расчетах; задание решено не полностью или в общем виде.
0-5	«неудовлетворительно»		Задание не выполнено или выполнено неправильно.

**Лист актуализации оценочных материалов по дисциплине  
«Технологии и методы программирования»**

Оценочные материалы пересмотрены,  
обсуждены и одобрены на заседании кафедры

Протокол от « 22 » мая 2018 г. № 10

Зав. кафедрой В. В. Ганзев В.С.

Оценочные материалы пересмотрены,  
обсуждены и одобрены на заседании кафедры

Протокол от « 20 » мая 2019 г. № 10

Зав. кафедрой В. В. Ганзев В.С.

Оценочные материалы пересмотрены,  
обсуждены и одобрены на заседании кафедры

Протокол от « 30 » июня 2020 г. № 12

Зав. кафедрой В. В. Ганзев В.С.

Оценочные материалы пересмотрены,  
обсуждены и одобрены на заседании кафедры

Протокол от « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ г. № \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_

---