

Министерство образования и науки Республики Дагестан
Государственное бюджетное профессиональное образовательное учреждение
Республики Дагестан
«Кизлярский профессионально-педагогический колледж»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

**по учебной дисциплине ОП.03 Основы алгоритмизации и
программирования**

Код и наименование специальности (профессии): 10.05.02 Обеспечение
информационной безопасности автоматизированных систем

Форма обучения

Кизляр, 2023г.

Фонд оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности/профессии СПО (10.05.02 Обеспечение информационной безопасности автоматизированных систем)

Разработчики:

Дамурова Ф.З., преподаватель ГБПОУ РД КППК

(место работы) (занимаемая должность) (инициалы, фамилия)

Рассмотрено и одобрено ПЦК профессиональных дисциплин по
техническим специальностям

Протокол № 1 от 29 08 2023 г.

Председатель ПЦК Раджабова А.Н. / А. 9J
(ФИО) (подпись)

СОДЕРЖАНИЕ

| | |
|--|----|
| 1. Общие положения | 4 |
| 3. Распределение оценивания результатов обучения по видам контроля..... | 5 |
| 4. Распределение типов контрольных заданий по элементам знаний и умений при текущем контроле | 6 |
| 5. Распределение типов и количества контрольных заданий по элементам знаний и умений, контролируемых на промежуточной аттестации | 7 |
| 6. Структура контрольных заданий..... | 8 |
| 6.1 Задания текущего контроля | 8 |
| 6.2 Задания промежуточной аттестации | 35 |
| 6.2.1 Текст задания | 35 |
| 6.2.2 Критерии оценки образовательных достижений..... | 46 |

1. Общие положения

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, осваивающих программу учебной дисциплины **ОП.03 Основы алгоритмизации и программирования**.

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

КОС разработаны в соответствии с:

- основной образовательной программой по специальности СПО 10.02.05 Обеспечение информационной безопасности автоматизированных систем
- программой учебной дисциплины ОП.03 Основы алгоритмизации и программирования

2. Результаты освоения дисциплины, подлежащие проверке

| Результаты обучения (освоенные умения, усвоенные знания) | Основные показатели оценки результатов |
|--|--|
| У 1 Работать в среде программирования | –использование средств редактора создания кода; –применение средств автоматизированной отладки и интерпретации результатов; –использование функций интегрированной среды |
| У 2 Использовать языки программирования высокого уровня | –владение синтаксисом языка; –семантически правильное использование программных конструкций; –разработка алгоритма и программы, созданной на языке программирования высокого уровня, для различных наборов тестовых данных |
| З 1 Типы данных | –перечисление основных типов данных; –описание типов данных; –воспроизведение зарезервированных слов для обозначения типов данных; –сравнительная характеристика типов данных языка программирования; –выполнение операций с различными типами данных |
| З 2 Базовые конструкции изучаемых языков программирования; | –характеристика классов языков программирования; –описание назначения базовых конструкций языка программирования; –описания синтаксиса базовых конструкций; –описания семантики базовых конструкций; распознавание алгоритмической конструкции по её характеристике. |

| | |
|---|---|
| 3 3 Интегрированные среды программирования на изучаемых языках. | <p><i>Знание:</i> описание основных компонент интегрированной среды программирования</p> <p><i>Понимание:</i> Объяснение применения основных компонент интегрированной среды программирования</p> |
|---|---|

3. Распределение оценивания результатов обучения по видам контроля

| Наименование элемента умений или знаний | Виды аттестации | |
|---|---|--------------------------|
| | Текущий контроль | Промежуточная аттестация |
| У 1 Работать в среде программирования | Формализованное наблюдение и оценка выполнения практической работы. Контрольная работа | экзамен |
| У 2 Использовать языки программирования высокого уровня | Формализованное наблюдение и оценка выполнения практической работы. Контрольная работа | экзамен |
| 3 1 Типы данных | Формализованное наблюдение и оценка выполнения практической работы. Контрольная работа | экзамен |
| 3 2 Базовые конструкции изучаемых языков программирования; | Формализованное наблюдение и оценка выполнения практической работы. Контрольная работа | экзамен |
| 3 3 Интегрированные среды программирования на изучаемых языках. | Формализованное наблюдение и оценка выполнения практической работы. Контрольная работа | экзамен |

4. Распределение типов контрольных заданий по элементам знаний и умений при текущем контроле

| Содержание учебного материала по программе УД | Тип контрольного задания | | | | |
|---|--------------------------|--------|--------|--------|--------|
| | У1 | У2 | З1 | З2 | З3 |
| Раздел 1 Основы алгоритмизации и программирования | | | | | |
| Тема 1.1. Основы алгоритмизации | ПР | ПР | ПР, У | ПР, У | ПР, У |
| Тема 1.2. Основы программирования | ПР | ПР | ПР, У | ПР, У | ПР, У |
| Раздел 2. Объектно-ориентированное программирование | | | | | |
| Тема 2.1. Объектно-ориентированная модель программирования | ПР, Т | ПР, Т | ПР, Т | ПР, Т | ПР, Т |
| Тема 2.2 Работа с файловой системой | ПР, КР | ПР, КР | ПР, КР | ПР, КР | ПР, КР |
| Раздел 3. Разработка WPF приложений | | | | | |
| Тема 3.1 Понятие WPF разработки. Основы разработки. | ПР | ПР | ПР | ПР | ПР |
| Тема 3.2 Разработка приложений | ПР | ПР | ПР, П | ПР, П | ПР, П |

У – устный ответ,

Т - тест,

П - проект,

КР – письменная контрольная работа,

ПР - практическая работа,

5. Распределение типов и количества контрольных заданий по элементам знаний и умений, контролируемых на промежуточной аттестации

| Содержание учебного материала по программе УД | Тип контрольного задания | | | | |
|---|--------------------------|-------|-------|-------|-------|
| | У1 | У2 | З1 | З2 | З3 |
| Раздел 1 Основы алгоритмизации и программирования | | | | | |
| Тема 1.1. Основы алгоритмизации | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |
| Тема 1.2. Основы программирования | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |
| Раздел 2. Объектно-ориентированное программирование | | | | | |
| Тема 2.1. Объектно-ориентированная модель программирования | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |
| Тема 2.2 Работа с файловой системой | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |
| Раздел 3. Разработка WPF приложений | | | | | |
| Тема 3.1 Понятие WPF разработки. Основы разработки. | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |
| Тема 3.2 Разработка приложений | ПР, У | ПР, У | ПР, У | ПР, У | ПР, У |

У – устный ответ

ПР – практическая работа

6. Структура контрольных заданий

6.1 Задания текущего контроля

Контрольная работа по теме «Массивы»

1. Найти сумму всех элементов одномерного массива целых чисел кратных числам 4,5 и 8. (1 балл)
2. Найти сумму номера чётных и нечетных элементов одномерного массива. (1 балл)
3. Найти сумму положительных и отрицательных элементов одномерного массива. (1 балл)
4. Найти количество положительных и отрицательных элементов одномерного массива. (1 балл)
5. Дан одномерный числовой массив $T(k)$. Вычислить сумму произведений всех троек соседних чисел. (2 балл)
6. Дан массив $B(n)$, содержащий большое количество нулевых элементов. Заменить все группы подряд встречающихся нулей на один ноль. (2 балла)
7. Дана последовательность целых чисел. Найти количество различных чисел в этой последовательности. (2 балла)
8. Дан массив целых чисел. Найти в этом массиве минимальный элемент m и максимальный элемент M . Получить в порядке возрастания все целые числа из интервала $(m; M)$, которые не входят в данный массив. (2 балла)
9. Найти минимальный элемент массива $A(m*n)$. (1 балл)
10. Найти максимальный элемент массива $A(m*n)$. (1 балл)
11. Найти сумму элементов массива $A(m*n)$ кратных заданному числу. (1 балл)
12. Найти количество положительных и отрицательных элементов в массиве
13. $A(m*n)$. (1 балл)
14. Вывести номера чётных элементов массива $A(m*n)$. (1 балл)
15. В данной целочисленной квадратной матрице $A(5,5)$ указать индексы всех элементов, имеющих наибольшее значение. (1 балл)
16. Дана квадратная целочисленная матрица $F(m,m)$. Найти суммы элементов тех строк, имеющих четные элементы на главной диагонали. (2 балла)
17. Для заданной матрицы размером 8 на 8 найти такие k , что k -я строка матрицы совпадает с k -м столбцом. (3 балла)
18. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент. (2 балла)
19. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине — в позиции (2,2), следующий по величине — в позиции (3,3) и т. д., заполнив таким образом всю главную диагональ. (4 балла)
20. Найти номер первой из строк, не содержащих ни одного положительного элемента. (3 балла)

11 баллов – «удовлетворительно»

15 баллов – «хорошо»

>19 баллов – «отлично»

Контрольная работа по теме «Методы»

1. В классе MyClass перед методом Main() создайте статическую процедуру с именем WriteString со строковым параметром input, которая будет выводить в консоль с новой строки значение input, а затем в методе Main() вызовите эту процедуру с параметром "Hello! It's procedure".

```
class MyClass
{
    static void Main(string[] args)
    {

    }
}
```

2. Что выведет на экран данный код?

```
class Program { static void DecrementArguments(int arg1, ref int arg2) { arg1--; --arg2; }
static void Main(string[] args)
{ int a = 5, b = 9; DecrementArguments(a, ref b);
Console.WriteLine(a + " " + b); Console.ReadKey(); }
}
```

3. Найдите и исправьте ошибки в коде

```
class Program
{
    static void Greetings(outt string s)
    {
        s = "Hello, everyone!";
        Console.WriteLine(s);
    }

    static void Main(string[] args)
    {
        string @string;
        Greetings(@string);
        Console.ReadKey();
    }
}
```

4. В класс MyClass поместите метод AddFive(), который будет принимать в качестве аргумента переменную d типа double и будет возвращать значение d, к которому прибавили 5.

```
namespace Program
{
    class MyClass
    {

    }
}
```

5. В тело класса MyMathClass поместите метод `ulong power(uint x, uint n)`, который с помощью любого цикла и приращения результата в цикле будет вычислять результат возведения `x` в степень `n`. При этом метод должен корректно возводить число в нулевую степень (то есть возвращать 1).
6. Найдите и исправьте ошибки в коде для правильной его работы

```
class Program
{
    static void RecursiveFunction(int n)
    {
        Console.WriteLine("Верхняя половина метода: {0}", n);

        if (n > 0)
            RecrsiveFunction n - 1);

        Console.WriteLine("Нижняя половина метода: {0}", n);
    }

    static void Main(string[] args)
    {
        RecursiveFunction(5);
        Console.ReadKey();
    }
}
```

Тест по теме «Объектно-ориентированное программирование»

1. Что неправильно в определении следующих интерфейсов:

```
public abstract interface IAction
{
    void Move();
}
internal interface IRunAction : IAction
{
    void Run();
}
```

- a) Интерфейсы не могут иметь модификатор доступа `internal`
 - b) Интерфейсы не могут иметь модификатор `abstract`
 - c) Интерфейс `IRunAction` должен определить метод `Move`, так как базовый интерфейс `IAction` является абстрактным
 - d) От абстрактного интерфейса нельзя наследоваться
2. Корректна ли следующая перегрузка методов? Если да, то почему? Если нет, то почему?

```
static void IncrementVal(ref int val)
{
    val++;
    Console.WriteLine(val);
}
```

```
static void IncrementVal(int val)
```

```
{
    val++;
    Console.WriteLine(val);
}
```

- a) Да
- b) Нет

3. Что будет выведено на консоль при выполнении следующей программы:

```
class Program
{
    int val = 8;
    static void Main(string[] args)
    {
        int val = 10;
        Doubled(ref val);
        Console.WriteLine(val);
        Console.ReadKey();
    }
}
```

```
static void Doubled(ref int val)
{
    val *= 2;
}
```

- a) 16
- b) 20
- c) 10
- d) Программа не скомпилируется

4. Какое значение имеет константа Sunday в следующем перечислении?

```
enum Day
{
    Monday = 5,
    Tuesday,
    Wednesday,
    Thursday = 14,
    Friday = Tuesday,
    Saturday,
    Sunday
}
```

- a) 2
- b) 14
- c) 6
- d) 8

5. Делегат в языке C# задает:

- a) класс функций с заданной сигнатурой
- b) функцию с заданной сигнатурой и с заданной реализацией
- c) класс, у которого все методы представляют собой функции

6. Ключевое слово interface в языке C# задает описание:

- a) пользовательского интерфейса класса
- b) открытой части класса
- c) частного случая класса
- d) абстрактного класса

7. Дано перечисление enum TC {red, blue, yellow} и три переменных c1, c2, c3 этого класса. В результате присваиваний: c1 = TC.yellow; c2 = TC.blue; c3 = c1 | c2; переменная c3 получит значение

- a) red
- b) blue
- c) yellow
- d) ни одно из этих значений

8. Пусть задано описание интерфейса и класса: interface IP{string M(string s); string M1(int s);} public class C1:IP{string IP.M(string s){return (s+s);} string IP.M1(int x){return x.ToString();}public int M (int s) { return (s++);} }. Какие из объявлений в клиентском классе выполнимы

- a) C1 it1 = new C1(); it1.M(7777)
- b) C1 it2 = new C1(); string s ="ss"; s =it2.IP.M(s)
- c) C1 it3 = new C1(); string s ="ss"; s =((IP)it3).M(s)
- d) IP it4 = new IP(); string s= "ss"; s = it4.M(s)
- e) IP it5 = (IP) new C1(); string s= "ss"; s = it5.M(s)

9. Метод можно описать на уровне:

- a) класса
- b) пространства имен
- c) проекта
- d) решения

10. Дано перечисление enum TC {red, blue, yellow} и три переменных c1, c2, c3 этого класса. В результате присваиваний: c1 = TC.yellow; c2 = TC.blue; c3 = c1 | c2; эти переменные получат следующие значения базового типа:

- a) 0, 1, 2
- b) 2, 1, 3
- c) 1, 2, 3
- d) 2, 1, 0

11. При вызове аргумента "по значению":

- a) формальный аргумент должен снабжаться ключевым словом ref
- b) создается копия фактического аргумента
- c) значение фактического аргумента не меняется в результате вызова
- d) фактический аргумент не может быть именем

12. Отметьте истинные высказывания:

- a) только процедуры являются методами класса
- b) формальный аргумент метода может быть выражением

- c) ключевые слова `ref` и `out` являются частью сигнатуры метода
 - d) перегруженными называются методы с одинаковыми именами, но с разной сигнатурой
13. Метод является:
- a) функцией, если тип возвращаемого значения отличен от `void`
 - b) функцией, если указан модификатор метода `function`
 - c) процедурой, если указан `void` как тип возвращаемого значения
 - d) процедурой, если указан модификатор метода `procedure`
14. Формальный аргумент метода должен сопровождаться ключевым словом:
- a) `ref` – если аргумент принадлежит к ссылочному типу
 - b) `out` – если аргумент является выходным параметром метода
 - c) `in` – если аргумент является входным параметром метода
15. Правда ли, что:
- a) функция всегда вычисляет значение, возвращаемое в качестве результата
 - b) методы родительского класса с модификатором `private` недоступны для вызова в методе, определяемом в классе потомка
 - c) число фактических аргументов при вызове метода должно совпадать с числом формальных аргументов этого метода
16. Правда ли, что:
- a) тело метода является блоком, представляющим последовательность операторов и описаний переменных, заключенную в фигурные скобки
 - b) функцию нельзя вызывать как оператор
 - c) типы формального и фактического аргумента должны быть идентичными
17. Отметьте верные утверждения:
- a) существование в классе методов с одним и тем же именем называется перегрузкой
 - b) любую функцию можно записать в виде процедуры
 - c) любую процедуру можно записать в виде функции
18. Отметьте истинные высказывания:
- a) использование цикла `For Each` при работе с массивом не допускается
 - b) класс `Array` является потомком класса `Object`
 - c) свойство `Length` и метод `GetLength` всегда возвращают один и тот же результат
 - d) если `S` и `T` массивы ссылочного типа, то присваивание `S=T` может быть допустимым
 - e) в результате присваивания `S=T`, где `S` и `T` массивы, создается копия массива `T`
19. Пусть процедура `ItWorks` имеет следующий заголовок:
`public void ItWorks(object[] A)`. Какой массив может быть передан ей в качестве аргумента?
- a) `int[] x`
 - b) `string[] s`
 - c) `Person[] p`; (`Person` – класс, определенный пользователем)
 - d) `Anketa[] a`; (`Anketa` – структура, определенная пользователем)
20. Класс имеет:
- a) только один конструктор
 - b) только конструктор по умолчанию
 - c) не более одного статического конструктора
 - d) не более одного закрытого конструктора
 - e) несколько конструкторов с одинаковой сигнатурой

21. Тело класса может содержать описание:
- a) полей
 - b) методов
 - c) классов
 - d) интерфейсов
 - e) пространств имен
 - f) событий
22. Отметьте истинные высказывания:
- a) любые методы класса могут вызывать статические методы
 - b) конструкторы класса вызываются автоматически
 - c) у класса может быть только индексатор с именем `this`, возможно перегруженный
 - d) класс не может иметь закрытых методов
23. Отметьте истинные высказывания:
- a) статическое поле класса доступно в любом методе класса
 - b) статический конструктор класса вызывается автоматически
 - c) у класса может быть задано несколько методов-свойств
 - d) у класса может быть задано несколько индексаторов с разными именами
 - e) для класса можно задать несколько операций
24. Поля класса:
- a) могут иметь любой из модификаторов доступа: `public`, `private`, `protected`, `internal`
 - b) могут иметь модификатор `static`
 - c) обычно закрываются
 - d) закрытые поля недоступны методам класса
25. Закрытые методы класса
- a) позволяют клиентам не вникать в детали реализации
 - b) составляют интерфейс класса
 - c) не могут быть вызваны клиентами
26. Правда ли, что:
- a) статические поля представляют общий информационный пул для всех объектов классов, позволяя извлекать и создавать общую информацию
 - b) конструктор не может быть объявлен с атрибутом `private`
 - c) для класса C# нельзя задать деструктор
27. Правда ли, что:
- a) для класса нельзя задать собственные константы
 - b) клиент не может получить прямой доступ к закрытым полям класса, но может получить доступ к ним, например через методы-свойства класса
 - c) независимо от значения атрибута доступа, все методы и поля класса доступны для не статических методов класса
28. Какие утверждения справедливы для развернутых и ссылочных типов
- a) структуры относятся к развернутым типам
 - b) классы относятся к ссылочным типам
 - c) в языке C# есть модификатор `expand`, позволяющий отнести класс к развернутому типу

d) объектам развернутого типа память отводится в куче

29. Особенности структур являются

- a) структуры не могут наследовать интерфейсы
- b) структуры могут наследовать интерфейсы
- c) поля структуры не могут быть инициализированы в момент объявления
- d) для структур нельзя создать собственный конструктор без аргументов

30. Отметьте истинные высказывания

- a) один класс может наследовать несколько интерфейсов
- b) один интерфейс может наследоваться несколькими классами
- c) из-за коллизии имен дублируемое наследование интерфейсов запрещено
- d) интерфейс может быть наследником нескольких интерфейсов

31. Какое число (или какие числа) выведет на консоль следующая программа и почему:

```
class Program
{
    static void Main(string[] args)
    {
        int i = 0;
        Console.Write(i++ + Calculate(i));
        Console.WriteLine(i);

        Console.ReadKey();
    }

    public static int Calculate(int i)
    {
        Console.Write(i++);
        return i;
    }
}
```

- a) 120
- b) 121
- c) 129

32. Почему не компилируется следующая программа?

```
class Example
{
    public static const double KOEF = 4.5;
    public readonly double field = 7.8;

    public Example(int f)
    {
        this.field = KOEF * f;
    }
}

class Program
{
    static void Main(string[] args)
```

```

{
    Example example = new Example(10);
    Console.WriteLine(example.field);

    Console.ReadKey();
}
}

```

- a) При определении константы нельзя использовать модификаторы доступа, например, модификатор `public`
- b) Константы с модификатором `public` могут располагаться только в классе, который тоже определен с модификатором `public`
- c) Константа не может быть статической (то есть не может быть определена с модификатором `static`)
- d) Значение поля для чтения может быть установлено только один раз, здесь же оно устанавливается два раза

33. Вам надо определить в классе переменную, которая должна быть доступна из любого места в текущем проекте. Какой модификатор (или модификаторы, если их несколько) вы будете использовать?

- a) `internal`
- b) `protected`
- c) `protected internal`
- d) `private`
- e) `private protected`
- f) `public`

34. Что будет выведено на консоль при выполнении следующей программы:

```

class Program
{
    int x = 8;
    static void Main(string[] args)
    {
        int x = 9;

        {
            x++;
        }

        Console.WriteLine(x);
        Console.ReadKey();
    }
}

```

- a) 9
- b) 10
- c) 1
- d) Программа не скомпилируется

35. Какое числовое значение имеет константа `Season.Summer` в следующем перечислении?

```

enum Season
{
    Winter = 250,
    Spring,

```


Summer,
Autumn

- a) Значение не определено
- b) 252
- c) 250

36. Интерфейс отличается от абстрактного класса тем, что:

- a) у абстрактного класса все методы абстрактны, а у интерфейса некоторые методы могут иметь реализацию
- b) у интерфейса все методы абстрактны, а у абстрактного класса некоторые методы могут иметь реализацию
- c) потомок может иметь в качестве родителя несколько интерфейсов, но только один абстрактный класс

37. Какие утверждения справедливы относительно взаимных преобразований структур в классы

- a) для преобразования структуры в класс достаточно заменить слово `struct` словом `class`
- b) для преобразования класса в структуру всегда достаточно заменить слово `class` словом `struct`
- c) для преобразования класса в структуру иногда достаточно заменить слово `class` словом `struct`
- d) для преобразования класса в структуру всегда необходимо изменить конструктор класса
- e) для преобразования класса в структуру иногда необходимо изменить конструктор класса

38. Правда ли, что:

- a) методы-свойства обеспечивают различные стратегии доступа к закрытым полям класса
- b) индексатор – это специальный инструментальный, нумерующий методы класса
- c) статическое поле с атрибутом `readonly` может использоваться в качестве константы класса

39. При определении собственных классов можно задать:

- a) только конструкторы с атрибутом `public`
- b) статический конструктор
- c) конструкторы с атрибутами `private` и `protected`

40. Метод-Свойство класса

- a) может иметь атрибут доступа `private`
- b) может включать методы `get` и `set`
- c) включает только один из методов `get` или `set`
- d) позволяет реализовать различные стратегии доступа к закрытым полям класса
- e) обязательно должно быть указано для каждого закрытого поля класса

41. Отметьте истинные высказывания:

- a) каждый объект содержит набор из всех полей, определенных в классе
- b) создание объекта выполняется операцией `new` с вызовом конструктора класса
- c) для реализации стратегии доступа к закрытым полям класса используются методы-свойства

42. Особенности структур являются

- a) структуры не могут наследовать интерфейсы
- b) структуры могут наследовать интерфейсы
- c) поля структуры не могут быть инициализированы в момент объявления

d) для структур нельзя создать собственный конструктор без аргументов

43. Отметьте истинные высказывания

- a) поля структуры не могут быть ссылками
- b) структура наследует методы класса `object`
- c) перечисление наследует методы класса `object`
- d) структуры и перечисления – частные случаи классов

44. При создании объектов перечислений

- a) нельзя вызывать конструктор
- b) требуется их явная инициализация
- c) объекты получают значение, даже если инициализация не задана

45. Для классов родителей и потомков справедливо следующее утверждение:

- a) у родительского класса может быть несколько непосредственных потомков
- b) у класса потомка может быть несколько непосредственных родительских классов
- c) класс родитель может быть сам себе родитель
- d) класс потомок может быть собственным потомком

46. Перечисление – это

- a) абстрактный класс
- b) класс без собственных методов
- c) класс без собственных полей

47. Для понятия «полиморфизм» справедливы следующие утверждения

- a) полиморфизм определяется для семейства классов, связанных отношением наследования
- b) полиморфизм определяется для семейства классов, связанных отношением вложенности
- c) полиморфизм означает, что в классе заданы перегруженные методы
- d) реализация полиморфизма построена на динамическом связывании
- e) реализация полиморфизма предполагает статический контроль типов

48. Отметьте истинное высказывание:

- a) класс называется абстрактным, если он не вводит собственных полей данных
- b) если метод класса `A` объявлен с модификатором `virtual`, то для него применяется статическое связывание
- c) у класса может быть несколько непосредственных родительских классов
- d) у класса может быть только один непосредственный потомок
- e) в проектах на `C#` контроль типов выполняется на этапе компиляции

49. Отметьте истинные высказывания:

- a) вызов конструктора приводит к вызову конструкторов всех предков класса
- b) клиенту доступны все методы поставщика
- c) наследнику доступны все методы предков
- d) абстрактный класс может иметь полностью реализованный метод

50. В родительском классе описан метод `public virtual void M(int x) {}`.

Какое объявление в классе потомке вызовет ошибку на этапе компиляции, не допускающую выполнения проекта?

- a) `override void M(int x){}`

- b) `public override void M(int item){ }`
 - c) `public new void M(int x){ }`
51. В классе `Math` определены методы (функции):
- a) `Sin`
 - b) `Tg`
 - c) `Ln`
 - d) `MaxValue`
 - e) `Min`
52. В каком фрагменте возникнет ошибка:
- a) `sbyte sb1=1, sb2= 2, sb3 = sb1+ sb2`
 - b) `ulong ul1 =1, ul2= 2, ul3 = ul1 + ul2`
 - c) `int x; if (5 >4) x= 4`
 - d) `int x=5, y; if(x>4) y=x`
53. В каких фрагментах возникнет ошибка:
- a) `double x; x=Math.Sin(1);`
 - b) `Math M1 = new Math(); int x; x= M1.Sin(1);`
 - c) `int x; x=Random.Next();`
 - d) `int x; Random R1 = new Random(); x=R1.Next();`
54. Чему будет равно `x` в результате выполнения следующего фрагмента: `int x=5; for(int i =1; i<5; i++); x+=5;`
- a) 5
 - b) 10
 - c) 25
 - d) 30
 - e) 20
55. Тело цикла `for (int i = 1, j=-10; j<10; i--) j++;`
- a) ни разу не выполнится
 - b) выполнится 3 раза
 - c) произойдет заикливание
 - d) выполнится конечное число раз, большее 3-х
56. В каких фрагментах возникнет ошибка компиляции:
- a) `int x=5, y =2*2+1, z; if (y=x) z=2*x; else z=x+y;`
 - b) `int a=2, b=3,c=5, x;if(a<b)if(c>a)if(c>a+b) x=a+b+c;`
 - c) `int x, y, z; if (5 >4) x=5; y=x; z=y;`
57. Оператор `return`
- a) обязателен в функциях
 - b) возможен в процедурах
 - c) завершает выполнение функции или процедуры

Тест « Основы XAML. Компоновка. Элементы управления.

Для заполнения теста использовать ссылку <https://forms.gle/iYBAkaQXq4gFZ2Bv8>

Вопрос 1. (выбрать 1 правильный вариант ответа)

Что будет выведено на экран?

```
using System.Collections.Generic;
namespace Test
{
    public partial class MainWindow
    {
        public MainWindow() { InitializeComponent(); }
        public IList<IMessage> Messages
        { get { return new List<IMessage> { new Error(), new Info() }; } }
    }
    public class IMessage { }
    public class Error : IMessage { }
    public class Info : IMessage { }
}

<Window x:Class="Test.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:test="clr-namespace:Test"
        x:Name="this">
    <Window.Resources>
        <DataTemplate DataType="{x:Type test:IMessage}">
            <StackPanel Orientation="Horizontal">
                <TextBlock Text="IMessage-"/><TextBlock Text="{Binding}" />
            </StackPanel>
        </DataTemplate>
    </Window.Resources>
    <ListBox ItemsSource="{Binding Messages, ElementName=this}" />
</Window>
```

- 1

Error
Info
- 2 Ничего не будет выведено
- 3

IMessage-Error
IMessage-Info
- 4

IMessage-Test.Error
IMessage-Test.Info
- 5

IMessage-
IMessage-

Вопрос 2. (возможен выбор нескольких вариантов ответа)

Для каких элементов управления свойство "SelectedItem" допускает "двунаправленный байндинг"?

- 1 DataGrid
- 2 ListView
- 3 ListBox
- 4 ComboBox
- 5 TreeView

Вопрос 3. (выбрать 1 правильный вариант ответа)

```
<Window x:Class="Test.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <StackPanel x:Name="panel" Height="200" Background="Red">
        <StackPanel.Triggers>
            <EventTrigger RoutedEvent="Button.Click" SourceName="button">
                <BeginStoryboard>
                    <Storyboard>
                        <DoubleAnimation Storyboard.TargetProperty="Height"
                                         Duration="0:0:1" From="100" To="200" />
                    </Storyboard>
                </BeginStoryboard>
            </EventTrigger>
        </StackPanel.Triggers>
        <Button x:Name="button">Expand</Button>
        <Button Click="OnCollapseButtonClick">Collapse</Button>
    </StackPanel>
</Window>

using System.Windows;
namespace Test
{
    public partial class MainWindow
    {
        public MainWindow() { InitializeComponent(); }
        public void OnCollapseButtonClick(object sender, RoutedEventArgs e)
        {
            panel.Height = 100;
        }
    }
}
```

Шаг 1. Нажимают кнопку "Collapse".

Шаг 2. Нажимают кнопку "Expand".

Шаг 3. Нажимают кнопку "Collapse".

Что будет происходить на экране?

1. Шаг 1. Произойдёт ошибка выполнения.
2. Шаг 1. Панель быстро уменьшится. Шаг 2. Произойдёт ошибка выполнения.
3. Шаг 1. Панель не изменит размер. Шаг 2. Панель быстро уменьшится и медленно увеличится. Шаг 3. Панель не изменит размер.
4. Шаг 1. Панель быстро уменьшится. Шаг 2. Панель медленно увеличится. Шаг 3. Панель не изменит размер.
5. Произойдёт ошибка компиляции.
6. Шаг 1. Панель быстро уменьшится. Шаг 2. Панель медленно увеличится. Шаг 3. Панель быстро уменьшится.
7. Шаг 1. Панель не изменит размер. Шаг 2. Произойдёт ошибка выполнения.

Вопрос 4. (возможен выбор нескольких вариантов ответа)

В каком потоке происходит обновление визуальных элементов в WPF? `<Window x:Class="MainWindow" xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"> <TextBox Text="{Binding Text}"></TextBox> </Window>` Если свойство `Text` у `ViewModel` обновляется следующим образом: `Task.Run(() => Text = "Hello, world");`

1. В новом `background` потоке
2. В `ThreadPool`-е
3. В потоке из `STA`
4. В потоке из `ThreadPool`-а

Вопрос 5. (возможен выбор нескольких вариантов ответа)

Какие ошибки содержатся в приведённом фрагменте XAML (укажите все)?

```
<TextBox>
  <TextBox.Text>
    <Binding Path="Value" ElementName="this"
      ConverterParameter="{Binding Flag}">
      <Binding.Converter>
        <test:MyConverter/>
      </Binding.Converter>
    </Binding>
  </TextBox.Text>
</TextBox>
```

1. Использование имени this в ElementName недопустимо.
2. ConverterParameter объявлен неверно.
3. Для Binding должен быть обязательно задан Mode.
4. Ошибок нет.
5. Binding задан отдельным тегом внутри Text, что недопустимо.
6. Объявление MyConverter должно содержать ключ, т.к. это ресурс.

Вопрос 6. (выбрать 1 правильный вариант ответа)

Выберите верное указание Margin такое, что левое поле равно 1 пикселю, правое – 2 пикселям, нижнее – 3 пикселям, верхнее – 4 пикселям:

1. Margin="1,2,3,4"
2. Margin="4,3,2,1"
3. Margin="1,2,4,3"
4. Margin="4,3,1,2"
5. Margin="1,4,2,3"

Вопрос 7. (выбрать 1 правильный вариант ответа)

В XAML имеется следующее объявление:

```
<TextBox x:Name="textBox" Text="Old Text"/>
```

Что произойдёт в результате выполнения следующего кода:

```
new Thread(o => textBox.Text = "New Text").Start();
```

- 1 Произойдёт ошибка выполнения.
 - Код не скомпилируется, т.к. поток создан неверно.
- 2 Код не скомпилируется, т.к. верный синтаксис `textBox.SetText("New Text")`.
 - Код не скомпилируется, т.к. необходимо установить `ApartmentState` в значение `MTA`.
- 3 Текст будет заменён на "New Text".
 -
- 4
- 5

Вопрос 8. (возможен выбор нескольких вариантов ответа)

Какие фрагменты XAML, позволяют вывести на экран "0.6-0.1"?

1.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sys="clr-namespace:System;assembly=mscorlib">
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding>
                    <Binding.Source>
                        <sys:Decimal>0.567</sys:Decimal>
                    </Binding.Source>
                </Binding>
                <Binding>
                    <Binding.Source>
                        <sys:Decimal>0.125</sys:Decimal>
                    </Binding.Source>
                </Binding>
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

2.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Window.Resources>
        <StringToDecimalConverter x:Key="myConverter"/>
    </Window.Resources>
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding Path="." Source="0.567"
                        Converter="{StaticResource myConverter}"/>
                <Binding Path="." Source="0.125"
                        Converter="{StaticResource myConverter}"/>
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

3.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <Window.Resources>
        <x:Decimal x:Key="number1">0.567</x:Decimal>
        <x:Decimal x:Key="number2">0.125</x:Decimal>
    </Window.Resources>
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding Source="{StaticResource number1}" />
                <Binding Source="{StaticResource number2}" />
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

4.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding Path="." Source="{x:Static 0.567m}" />
                <Binding Path="." Source="{x:Static 0.125m}" />
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

5.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:sys="clr-namespace:System;assembly=mscorlib">
    <Window.Resources>
        <sys:Decimal x:Key="number1">0.567</sys:Decimal>
        <sys:Decimal x:Key="number2">0.125</sys:Decimal>
    </Window.Resources>
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding Source="{StaticResource number1}" />
                <Binding Source="{StaticResource number2}" />
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

6.

```
<Window x:Class="Test.TestWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
    <TextBlock>
        <TextBlock.Text>
            <MultiBinding StringFormat="{0:F1}-{1:F1}">
                <Binding Path="." Source="0.567" />
                <Binding Path="." Source="0.125" />
            </MultiBinding>
        </TextBlock.Text>
    </TextBlock>
</Window>
```

Вопрос 9. (возможен выбор нескольких вариантов ответа)

```
<Canvas Width="200" Height="200">
  <Rectangle Width="50" Height="50" Fill="Green">
    ...
  </Rectangle>
</Canvas>
```

Что можно вставить вместо многоточия, чтобы сместить прямоугольник на 50 пикселей вправо?

1.

```
<Rectangle.RenderTransform>
  <MoveTransform X="50"/>
</Rectangle.RenderTransform>
```
2.

```
<Rectangle.Margin.Left>50</Rectangle.Margin.Left>
```
3.

```
<Rectangle.RenderTransform>
  <TranslateTransform X="50"/>
</Rectangle.RenderTransform>
```
4.

```
<Rectangle.Transformations>
  <ShiftTransformation X="50"/>
</Rectangle.Transformations>
```
5.

```
<Canvas.Left>50</Canvas.Left>
```
6.

```
<Rectangle.Transformations>
  <MoveTransformation X="50"/>
</Rectangle.Transformations>
```
7.

```
<Rectangle.Margin>
  <Left>50</Left>
</Rectangle.Margin>
```
8.

```
<Rectangle.Left>50</Rectangle.Left>
```

Вопрос 10. (возможен выбор нескольких вариантов ответа)

```
<Grid Width="100" Height="100">  
  <Grid.RowDefinitions>  
    <RowDefinition Height="Auto"/>  
  </Grid.RowDefinitions>  
  <TextBox VerticalAlignment="Stretch" AcceptsReturn="True"/>  
</Grid>
```

Необходимо сделать так, чтобы вертикальная полоса прокрутки появлялась только тогда, когда высота набранного текста станет больше высоты таблицы. Что нужно исправить?

1. Убрать VerticalAlignment="Stretch".
2. Добавить VerticalScrollBarVisibility="Auto" к атрибутам Grid.
3. Добавить VerticalScrollBarVisibility="Auto" к атрибутам RowDefinition.
4. Исправить RowDefinition.Height на *.
5. Добавить Grid.Row="0" к атрибутам TextBox.
6. Исправить значение VerticalAlignment на Scroll.
7. Добавить VerticalScrollBarVisibility="Auto" к атрибутам TextBox.
8. Добавить TextWrapping="Wrap" к атрибутам TextBox.

Вопрос 11. (возможен выбор нескольких вариантов ответа)

Известно, что Value равно 123.45678. Какие варианты ответов, выведут на экран 123.46?

1. `<TextBlock Text="{Binding Value, StringFormat=0.00}"/>`
2. `<TextBlock Text="{Binding Value, StringFormat={0.00}}"/>`
3. `<TextBlock Text="{Binding Value, StringFormat='{0:0.00}}"/>`
4. `<TextBlock Text="{Binding Value, StringFormat=\"0:0.00\"}"/>`
5. `<TextBlock Text="{Binding Value, StringFormat=\{0.00\}}"/>`
6. `<TextBlock Text="{Binding Value, StringFormat=0:0.00}"/>`

Вопрос 12. (возможен выбор нескольких вариантов ответа)

Какие из приведённых фрагментов XAML выполняются без ошибок?

1.

```
<TextBox>
  <TextBox.Text>
    <PriorityBinding>
      <Binding Path="MyText"/>
      <Binding Path="." Source="Empty String"/>
    </PriorityBinding>
  </TextBox.Text>
</TextBox>
```
2.

```
<TextBox x:Name="myTextBox">
  <TextBox.Style>
    <Style TargetType="{x:Type TextBox}">
      <Setter Property="Text" Value="{Binding MyText}"/>
      <Style.Triggers>
        <DataTrigger Binding="{Binding MyText}" Value="{x:Null}">
          <Setter Property="Text" Value="Empty String"/>
        </DataTrigger>
      </Style.Triggers>
    </Style>
  </TextBox.Style>
</TextBox>
```
3.

```
<TextBox Text="{Binding MyText, Converter={x:Static NullValueConverter}}"/>
```
4.

```
<TextBox Text="{Binding MyText, TargetNullValue=Empty String}"/>
```
5.

```
<TextBox Text="{Binding MyText, DataSourceNullValue=Empty String}"/>
```
6.

```
<TextBox Text="{Binding MyText, NullValue=Empty String}"/>
```
7.

```
<TextBox Text="{Binding MyText, SourceNullValue=Empty String}"/>
```
8.

```
<TextBox Text="{Binding MyText, FallbackValue=Empty String}"/>
```


Вопрос 13. (возможен выбор нескольких вариантов ответа)

Из приведённых фрагментов XAML выберите те, что позволят выводить на экран текст "MouseOver" при наведении курсора мыши на красный квадрат и скрывать, когда курсор мыши находится вне квадрата.

1.

```
<ControlTemplate>
  <StackPanel>
    <TextBlock x:Name="indicator"/>
    <Rectangle x:Name="rect" Width="100" Height="100" Fill="Red"/>
  </StackPanel>
  <ControlTemplate.Triggers>
    <Trigger SourceName="rect" Property="IsMouseOver" Value="True">
      <Setter TargetName="indicator"
        Property="TextBlock.Text"
        Value="MouseOver"/>
    </Trigger>
  </ControlTemplate.Triggers>
</ControlTemplate>
```

2.

```
<StackPanel>
  <TextBlock x:Name="indicator"/>
  <Rectangle x:Name="rect" Width="100" Height="100" Fill="Red">
    <Rectangle.Triggers>
      <EventTrigger RoutedEvent="Rectangle.MouseEnter">
        <Setter TargetName="indicator"
          Property="TextBlock.Text"
          Value="MouseOver"/>
      </EventTrigger>
      <EventTrigger RoutedEvent="Rectangle.MouseLeave">
        <Setter TargetName="indicator"
          Property="TextBlock.Text"
          Value=""/>
      </EventTrigger>
    </Rectangle.Triggers>
  </Rectangle>
</StackPanel>
```

3.

```
<StackPanel>
  <TextBlock x:Name="indicator"/>
  <Rectangle x:Name="rect" Width="100" Height="100" Fill="Red">
    <Rectangle.Triggers>
      <EventTrigger RoutedEvent="Rectangle.MouseEnter">
        <EventTrigger.Actions>
          <BeginStoryboard>
            <Storyboard>
              <StringAnimationUsingKeyFrames
                Storyboard.TargetName="indicator"
                Storyboard.TargetProperty="(TextBlock.Text)"
                Duration="0:0:0.1" FillBehavior="HoldEnd">
                <DiscreteStringKeyFrame Value="MouseOver"
                  KeyTime="0:0:0.1" />
              </StringAnimationUsingKeyFrames>
            </Storyboard>
          </BeginStoryboard>
        </EventTrigger.Actions>
      </EventTrigger>
      <EventTrigger RoutedEvent="Rectangle.MouseLeave">
        <EventTrigger.Actions>
          <BeginStoryboard>
            <Storyboard>
              <StringAnimationUsingKeyFrames
                Storyboard.TargetName="indicator"
                Storyboard.TargetProperty="(TextBlock.Text)"
                Duration="0:0:0.1" FillBehavior="HoldEnd">
                <DiscreteStringKeyFrame Value=""
                  KeyTime="0:0:0.1" />
              </StringAnimationUsingKeyFrames>
            </Storyboard>
          </BeginStoryboard>
        </EventTrigger.Actions>
      </EventTrigger>
    </Rectangle.Triggers>
  </Rectangle>
</StackPanel>
```

4.

```

<StackPanel>
  <TextBlock x:Name="indicator"/>
  <Rectangle x:Name="rect" Width="100" Height="100" Fill="Red">
    <Rectangle.Style>
      <Style>
        <Style.Triggers>
          <Trigger Property="UIElement.IsMouseOver" Value="True">
            <Setter TargetName="indicator"
              Property="TextBlock.Text"
              Value="MouseOver"/>
          </Trigger>
        </Style.Triggers>
      </Style>
    </Rectangle.Style>
  </Rectangle>
</StackPanel>

```

5.

```

<StackPanel>
  <TextBlock x:Name="indicator">
    <TextBlock.Style>
      <Style>
        <Style.Triggers>
          <DataTrigger
            Binding="{Binding IsMouseOver, ElementName=rect}"
            Value="True">
            <Setter Property="TextBlock.Text" Value="MouseOver"/>
          </DataTrigger>
        </Style.Triggers>
      </Style>
    </TextBlock.Style>
  </TextBlock>
  <Rectangle x:Name="rect" Width="100" Height="100" Fill="Red"/>
</StackPanel>

```

Критерии оценки: (для теста)

| Процент результативности (правильных ответов) | Оценка уровня подготовки | |
|---|--------------------------|---------------------|
| | балл (отметка) | вербальный аналог |
| 90 - 100 | 5 | отлично |
| 75 - 89 | 4 | хорошо |
| 60 - 74 | 3 | удовлетворительно |
| менее 60 | 2 | неудовлетворительно |

6.2 Задания промежуточной аттестации

6.2.1 Текст задания

ВОПРОСЫ К ЭКЗАМЕНУ (2 КУРС, 2 СЕМЕСТР)

1. Понятие алгоритма, свойства, виды алгоритмов.
2. Способы изображения алгоритмов. Графическое изображение алгоритма.
3. Основные этапы решения задачи на ЭВМ.
4. Язык C# и платформа .NET
5. Переменные и литералы в C#. Использование суффиксов.
6. Типы данных языка C#.
7. Арифметические операторы языка C#.
8. Порядные операции языка C#.

9. Операторы присваивания.
10. Преобразования базовых типов данных и управляющие символы.
11. Логические операторы.
12. Условные выражения языка C#.
13. Условные конструкции языка C#.
14. Тернарный оператор.
15. Цикл For. Операторы continue и break.
16. Цикл while, do...while Операторы continue и break.
17. Обработка исключений и использование меток.
18. Цикл по коллекции foreach.
19. Ссылочные типы данных.
20. Массивы: объявление, инициализация, использование в программе.
21. Многомерные массивы: объявление, инициализация, использование в программе.
22. Алгоритм сортировки массива (метод пузырька).
23. Работа с текстом в C#.
24. Текстовые файлы. Классы и методы для работы с текстовыми файлами.
25. Работа с консолью и класс Console.
26. Методы: определение, правила записи. Вызов метода.
27. Метод-процедура, метод-функция, механизм передачи аргументов, правило триединого соответствия.
28. Перегрузка статических методов.
29. Массив как аргумент метода, массив как результат метода.
30. Методы в библиотеке классов.
31. Базовые принципы ООП: инкапсуляция, наследование, полиморфизм.
32. Понятие класса и объекта.
33. Общая форма определения класса.
34. Характеристика объекта: поля, свойства, методы, события.
35. Модификаторы доступа к элементам класса.
36. Конструкторы, инициализация объектов.
37. Деструктор.
38. Инкапсуляция как управление доступом к данным. Свойство класса.
39. Наследование и полиморфизм. Иерархия классов: понятие и преимущество. Синтаксис наследования.
40. Интерфейсы: назначение и правила написания.
41. Способы реализации интерфейсов.
42. Наследование в интерфейсах.
43. Назначение и синтаксис структуры. Элементы тела структуры.
44. Тип данных enum: объявление и использование.
45. Операции с перечислениями.
46. Понятие коллекции. Класс List. Работа с ArrayList.

ЗАДАЧИ К ЭКЗАМЕНУ (2 КУРС, 2 СЕМЕСТР)

Задача 1. Определить функцию (статический метод) для вычисления наибольшего общего делителя двух целых натуральных чисел. В основной программе, используя функцию, сократить неотрицательную обыкновенную дробь. Дробь вводится с клавиатуры в виде неотрицательного числителя и положительного знаменателя.

Задача 2. Создайте приложение, которое читает и выводит информацию в текстовый файл, а также подсчитывает количество четных чисел в файле.

Задача 3. Описать переменную служащий, состоящую из имени, фамилии, отчества служащего, даты рождения, образования, домашнего адреса, профессии. Определить имена людей с высшим образованием. Выдать данные о служащем, который имеет ту или иную профессию.

Задача 4. Создайте приложение, позволяющее сохранять логины и пароли на внешнем носителе (с указанием даты создания и названия источника аутентификации).

Задача 5. Задан класс Book, который описывает книгу. Класс содержит следующие элементы:

- название книги;
- фамилия и имя автора;
- стоимость книги.

В классе Book нужно реализовать следующие методы: конструктор с 3 параметрами; метод Print(), который выводит информацию о книге.

Задача 6. В таблице хранятся следующие данные об учениках: фамилия, имя, отчество, рост, масса. Вычислить средний рост учеников, рост самого высокого и самого низкого ученика. Сколько учеников могут заниматься в баскетбольной секции, если рост баскетболиста должен быть больше 170 см?

Задача 7. Описать класс «Комната». Поля класса: длина и ширина комнаты, высота потолков, высота, ширина окна и двери. Описать методы класса:

- вычисление площади комнаты,
- ремонт: определить количество краски, необходимой для покраски стен (з расчёта 1 л на 2 3 кв. метра стены)

Задача 8. Ввести три целых числа, вывести их в порядке возрастания значений. Решение реализовать с применением двух методов-процедур. Первый метод упорядочивает значения двух параметров, а второй использует первый метод для упорядочивания трех параметров.

Задача 9. Напишите программы со статическим методом, аргументом которому передается двумерный массив. У метода, кроме аргумента-массива, есть еще два неинициализированных аргумента, результатов метод возвращает значение наибольшего элемента в массиве. Неинициализированным аргументам присваиваются значения индексов этого элемента.

Задача 10. Разработать библиотечные методы для создания одномерных и двумерных массивов со случайными значениями элементов.

Задача 11. Составить программу назначения стипендии студентам по результатам сессии, используя следующие правила: 1) если все оценки 5, назначается повышенная стипендия; 2) если все оценки 4 и 5, назначается обычная стипендия; 3) если есть оценка 3, стипендия не назначается. В результате работы программы должен быть напечатан список группы с оценками и средним баллом каждого студента и два списка фамилий (назначенных на повышенную и обычную стипендию).

Задача 12. Создайте приложение, которое читает и выводит информацию в текстовый файл, а также подсчитывает произведение чисел в файле.

Задача 13. Вычислить $\sin x$, используя приближенную формулу: где m – номер очередного члена, начиная с $m = 0$. Вычисление суммы ряда выполнить с использованием 10-ти членов ряда. В основной программе вводить значения x и выводить соответствующие значения синуса.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} \dots (-1)^m \frac{x^{2m+1}}{(2m+1)!} \dots,$$

Задача 14. Создайте приложение, которое читает и выводит информацию в текстовый файл, а также подсчитывает сумму двузначных чисел.

Задача 15. Создать базовый класс Покупатель: Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета; Конструктор; Методы: установка значений атрибутов, получение значений атрибутов, вывод информации. Создать массив объектов данного класса. Вывести список покупателей в алфавитном порядке и список покупателей, у которых номер кредитной карточки находится в заданном диапазоне.

Задача 16. Дан текстовый файл. Удалить из него все лишние пробелы, оставив между словами не более одного пробела. Результат поместить в новый файл.

Задача 17. Создать базовый класс Student, который будет содержать информацию о студенте (фамилия, курс обучения, номер зачетной книги). С помощью механизма наследования реализовать класс Aspirant (аспирант – студент, который готовится к защите кандидатской диссертации). Класс Aspirant есть производным от класса Student. В классах Student и Aspirant необходимо реализовать следующие элементы:

- конструкторы классов с соответствующим количеством параметров. В классе Aspirant для доступа к методам класса Student нужно использовать ключевое слово base;
- метод Print(), который выводит информацию о содержимом полей класса.

Задача 18. Дан файл, содержащий произвольный текст. Проверить, правильно ли в нем представлены круглые скобки (т.е. находится ли правее каждой открывающейся скобки закрывающаяся и левее закрывающейся — открывающаяся).

ВОПРОСЫ К ЭКЗАМЕНУ (3 КУРС, 1 СЕМЕСТР)

1. Работа с потоками и файловой системой. Работа с дисками
2. Работа с потоками и файловой системой. Работа с каталогами.
3. Работа с потоками и файловой системой. Работа с файлами. Классы File и FileInfo.
4. Работа с потоками и файловой системой FileStream. Чтение и запись файла.
5. Работа с потоками и файловой системой Чтение и запись текстовых файлов. StreamReader и StreamWriter.
6. Особенности платформы WPF.
7. Язык XAML. Пространства имен из C# в XAML.
8. Компоновка. Grid. GridSplitter.
9. Компоновка. StackPanel. DockPanel.
10. Компоновка. WrapPanel. Canvas.
11. Свойства компоновки элементов.
12. Элементы управления. Кнопки. CheckBox и RadioButton.
13. Элементы управления. Всплывающие подсказки ToolTip и Popup.
14. Элементы управления. Контейнеры GroupBox и Expander.
15. Элементы управления. ScrollView. Создание прокрутки.
16. Элементы управления. Текстовые элементы управления.

17. Элементы управления списками. ListBox, ComboBox, ListView.
18. Создание вкладок и TabControl, Меню.
19. Элементы управления. ToolBar. TreeView.
20. Элементы управления. DataGrid.
21. Элементы управления. ProgressBar и Slider.
22. Работа с датами. Calendar и DatePicker.
23. Работа с изображениями. Image и InkCanvas.
24. Модель событий в WPF. Маршрутизация событий.
25. Модель событий в WPF. События клавиатуры.
26. Модель событий в WPF. События мыши и фокуса.
27. Команды.
28. Ресурсы. Статические и динамические ресурсы.
29. Ресурсы. Словари ресурсов.
30. Стили.
31. Взаимодействие с базой данных.
32. Окна. Класс Window. Взаимодействие между окнами.
33. Диалоговые окна.

ЗАДАЧИ К ЭКЗАМЕНУ (3 КУРС, 1 СЕМЕСТР)

Задание 1: Разработать приложение «Калькулятор для расчёта обоев». При нажатии на кнопку «Рассчитать», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.

Задание 2: Разработать приложение «Калькулятор для расчёта краски». При нажатии на кнопку «Рассчитать», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.

Параметры комнаты

Ширину поверхности

4.3

м

Высоту (длину) поверхности

3.2

м

Параметры краски

Расход краски кв.м / л

5

м²/л

Количество слоев

1

шт.

Объем банки

3

л

Запас

☒ 10% запаса

☐ 15% запаса

☐ Без запаса

Рассчитать

Задание 3: Разработать приложение «Ипотечный калькулятор». При нажатии на кнопку «Рассчитать», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.

Укажите параметры кредита

Стоимость квартиры

2 000 000

руб.

Первоначальный взнос

500 000

руб.

или

%

Ставка кредита

12

% годовых в

руб.

Срок кредитования

15

лет

Рассчитать >>

Задание 4: Разработать приложение «Регистрация». При нажатии на кнопку «Регистрация», данные с формы должны сохраняться в текстовый файл.

Регистрация

Имя

Имя

Фамилия

Фамилия

E-mail

@

name@example.com

Пароль

••••••••••

Пароль повторно

••••••••

Страна

Россия

☐

Получать новости

Регистрация

Задание 5: Разработать приложение «Калькулятор расчёта плитки». При нажатии на кнопку «Посчитать», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.

Калькулятор расчета плитки

Введите размеры в метрах *

Длина пола

длина пола (стены), метров *(например, 5.3)*

Ширина пола

ширина пола (стены), метров *(например, 2.5)*

Введите размеры одной плитки *

Длина 1 плитки

длина, см

Ширина 1 плитки

ширина, см

Введите стоимость плитки (за 1 кв. м.)

Цена 1 кв. м. плитки

цена, рублей/кв. метр


ПОСЧИТАТЬ

Задание 6: Разработать приложение «Калькулятор для расчёта дробей». При нажатии на кнопку «Вычислить», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.


Задание 7: Разработать приложение «Калькулятор суточной нормы калорий и ИМТ». При нажатии на кнопку «Рассчитать», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением. Расчет нормы калорий происходит по формуле Миффлина-Сан Жеора. Базовый метаболизм рассчитывается по формуле $BMR = [9.99 * \text{вес (кг)}] + [6.25 * \text{рост (см)}] - [4.92 * \text{возраст (в годах)}] + 5$ (для мужчин) $- 161$ (для женщин). Умножив базовый метаболизм на коэффициент нагрузки получаем нужно количество калорий в сутки.

Коэффициенты нагрузки:


- Минимальная (1.2): сидячая работа, отсутствие спорта;
- Легкая (1.35): легкие физические упражнения около 3 раз за неделю, ежедневная утренняя зарядка, пешие прогулки;
- Средняя (1.55): спорт до 5 раз за неделю;
- Высокая (1.75): активный образ жизни вкупе с ежедневными интенсивными тренировками;
- Экстремальная (1.95): максимальная активность - спортивный образ жизни, тяжелый физический труд, длительные тяжелые тренировки каждый день.



Женщина



Мужчина



180

 Рост, см

80

 Вес, кг

Возраст:

30

 лет

 Активность:

средняя

?

Рассчитать

Задание 8: Разработать приложение «Калькулятор расчета заработной платы». При нажатии на кнопку «Рассчитать зарплату», данные с формы должны сохраняться в текстовый файл с итоговым расчетным значением.

Формулы для расчета заработной платы: 1) Полная ЗП = ОКЛАД * Р.КОЭФФ * ОТР ДНИ / РАБ ДНИ + ПРЕМИЯ 2) НДФЛ = Полная ЗП * 0.13 3) Зарплата на руки = Полная ЗП – НДФЛ

Центр Проектирования
☰

Калькулятор расчета заработной платы

| | |
|--|---|
| Оклад, руб = | |
| Премия, руб = | |
| Районный коэффициент (если есть) = | |
| Количество рабочих дней в месяце = | |
| Количество отработанных дней в месяце = | |
| <div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">Пример</div> | <div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">Рассчитать зарплату</div> |
| Полная зарплата, руб = | |
| НДФЛ 13%, руб = | |
| Зарплата на руки, руб = | |

При нажатии на кнопку Пример, загружаются данные с расчетом.

Центр Проектирования
☰

Калькулятор расчета заработной платы

| | |
|--|---|
| Оклад, руб = | 10000 |
| Премия, руб = | 1000 |
| Районный коэффициент (если есть) = | 1 |
| Количество рабочих дней в месяце = | 20 |
| Количество отработанных дней в месяце = | 20 |
| <div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">Пример</div> | <div style="border: 1px solid #ccc; padding: 2px 5px; display: inline-block;">Рассчитать зарплату</div> |
| Полная зарплата, руб = | 11000.00 |
| НДФЛ 13%, руб = | 1430.00 |
| Зарплата на руки, руб = | 9570.00 |

Задание 9: Разработать приложение «Калькулятор расчета услуги» По нажатию на кнопку *Рассчитать* происходит расчет услуг парикмахерской и на форму выводится *Стоимость за услуги, Сдача клиенту*. Напротив названия услуги ставим признак, что услуга была получена. Цена за одну Стрижку составляет 370 руб., стоимость Покраски равна 950 руб., стоимость Укладки составляет 560 руб. Сумма заказа рассчитывается автоматически, поле *Итого* недоступно для изменения, так же рассчитывается и поле *Сдача* при нажатии на кнопку *Рассчитать*. *Сумма принято* и *Скидка* вводится с клавиатуры.

По нажатию на кнопку *Оформить чек*: формируется текстовый документ (прилагается квитанция), который сохраняется *Уникальный номер чека_дата_стоимость*. Сформируйте чек для каждого нового заказа.

ООО "Соло"
 Добро пожаловать
 ККМ 00075411 #3969
 ИНН 1087746942040
 ЭКЛЗ 3851495566
 Чек №
27.04.20 00:30 СИС.

| Состав заказа | | |
|---------------|--|-------|
| Стрижка | | 370 |
| Покраска | | 950 |
| Итого | | =1320 |
| Внесено | | =1400 |
| Скидка (руб) | | =0 |
| Сдача | | =80 |

Задание 10: Разработать приложение «Калькулятор расчета заказа» По нажатию на кнопку *Рассчитать заказ* происходит расчет заказа в кафе и на форму выводится *Стоимость заказа, Сдача клиенту*. Напротив названия блюда вводим количество порций. Цена за одну порцию салата 45 руб., стоимость второго блюда равна 115 руб. *Сумма заказа* рассчитывается автома-

тически, недоступно для изменения, так же рассчитывается и поле Сдача при нажатии на кнопку *Рассчитать заказ*. Сумма принято вводится с клавиатуры.

По нажатию на кнопку *Оформить чек*: формируется текстовый документ (прилагается квитанция), который сохраняется *Уникальный номер чека_дата_стоимость*. Сформируйте чек для каждого нового заказа.

ООО "Кафе Встреч"
 Добро пожаловать
 ККМ 00075411 #3969
 ИНН 1087746942040
 ЭКЛЗ 3851495566
 Чек №
27.04.20 00:30 СИС.

Состав заказа

| | | |
|-----------------|--|-------|
| Салат «Оливье» | | 90 |
| Пюре с котлетой | | 230 |
| Итого | | =320 |
| Внесено | | =1000 |
| Сдача | | =680 |

00003751# 059705

6.2.2 Критерии оценки образовательных достижений

Для практической работы :

Отметка «5» ставится, если

Работа выполнена в полном объеме с соблюдением необходимой последовательности. Учащиеся работают полностью самостоятельно: подбирают необходимые для выполнения предлагаемых работ источники знаний, показывают необходимые для проведения работы теоретические знания, практические умения и навыки. Работа оформляется аккуратно, в наиболее оптимальной для фиксации результатов форме.

Отметка «4» ставится, если

Работа выполнена учащимся в полном объеме и самостоятельно. Допускаются отклонения от необходимой последовательности выполнения, не влияющие на правильность конечного результата. Учащийся использует, указанные учителем источники знаний, работа показывает знание учащимся основного теоретического материала и овладение умениями, необходимыми для самостоятельного выполнения работы. Могут быть неточности и небрежность в оформлении результатов работы.

Отметка «3» ставится, если

Работа выполняется и оформляется учащимся при помощи преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу учащихся. На выполнение работы затрачивается много времени. Учащийся показывает знания теоретического материала, но испытывает затруднение при самостоятельной работе с источниками знаний или приборами.