

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Романчук Иван Сергеевич
Должность: Ректор
Дата подписания: 16.02.2023 08:57:28
Уникальный программный ключ:
e68634da050325a9234284110b46018b7881130

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «Тюменский государственный университет»
Тюменский педагогический институт им. Д.И.Менделеева (филиал)
Тюменского государственного университета



УТВЕРЖДАЮ
Заместитель директора филиала
Шитиков П.М. Шитиков П.М.
«02» 02 2023 год

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
для обучающихся по программе подготовки специалистов среднего звена
09.02.07 Информационные системы и программирование
форма обучения очная

Зыбина Н.В. Основы алгоритмизации и программирования. Фонд оценочных средств дисциплины для обучающихся по программе подготовки специалистов среднего звена 09.02.07 Информационные системы и программирование. Форма обучения – очная. Тобольск, 2023.

Фонд оценочных средств дисциплины разработаны на основе ФГОС СПО по специальности 09.02.07 Информационные системы и программирование, утвержденного приказом Министерства образования и науки Российской Федерации от 09 декабря 2016 года, № 1547.

Фонд оценочных средств дисциплины опубликован на сайте ТюмГУ: Основы алгоритмизации и программирования. [электронный ресурс] / Режим доступа: <https://www.utmn.ru/sveden/education/#>

Содержание

- | | |
|---|---|
| 1. Общая характеристика фонда оценочных средств | 3 |
| 2. Паспорт фонда оценочных средств | 5 |
| 3. Типовые задания для оценки освоения учебной дисциплины | 9 |

1. ОБЩАЯ ХАРАКТЕРИСТИКА ФОНДОВ ОЦЕНОЧНЫХ СРЕДСТВ

1.1. Область применения программы

Фонд оценочных средств учебной дисциплины «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07 Информационные системы и программирование.

1.2. Место дисциплины в структуре программы подготовки специалистов среднего звена
Дисциплина входит в Общепрофессиональный цикл учебного плана специальности.

1.3. Цели и задачи дисциплины – требования к результатам освоения дисциплины.

В результате освоения дисциплины обучающийся должен обладать следующей компетенцией:

ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 11.1. Осуществлять сбор, обработку и анализ информации для проектирования баз данных.

ПК 11.2. Проектировать базу данных на основе анализа предметной области.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

Код ПК, ОК	Умения	Знания
ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	<ul style="list-style-type: none">– Разрабатывать алгоритмы для конкретных задач.– Использовать программы для графического отображения алгоритмов.– Определять сложность работы алгоритмов.– Работать в среде программирования.– Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.– Оформлять код программы в соответствии со стандартом кодирования.– Выполнять проверку, отладку кода программы.	<ul style="list-style-type: none">– Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.– Эволюцию языков программирования, их классификацию, понятие системы программирования.– Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.– Подпрограммы, составление библиотек подпрограмм– Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция

		и полиморфизма, наследования и перепределения
--	--	--

2.ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

п/п	Темы дисциплины, МДК, разделы (этапы) практики, в ходе текущего контроля, вид промежуточной аттестации с указанием семестра	Код контролируемой компетенции (или её части), знаний, умений	Наименование оценочного средства (с указанием количество вариантов, заданий и т.п.)
1.	Раздел 1. Основные принципы алгоритмизации и программирования	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Тест (8 вопросов) Online тест (40 вопросов)
2.	Раздел 2. Основы программирования на языке С#	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Тест (2 варианта) Коллоквиум (17 вопросов) Практическая работа (15 задач)
3.	Раздел 3 Модульное программирование	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Коллоквиум (вопросов 5) Практическая работа (14 задач)
4.	Раздел 4. Объектно-ориентированное программирование	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Коллоквиум (вопросов 35) Online тест (20 вопросов)
5.	Промежуточная аттестация в форме контрольной работы – 2 семестр	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Контрольная работа (2 варианта)
6.	Промежуточная аттестация в форме комплексного дифференцированного зачета– 3 семестр	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.	Вопросы к дифференцированному зачету (45 вопросов)

3. ТИПОВЫЕ ЗАДАНИЯ ДЛЯ ОЦЕНКИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Раздел 1. Основные принципы алгоритмизации и программирования	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
---	--

Тест «Основы понятия алгоритмизации»

1. По линии прямой связи передаются

- а) команды управления и информация об объекте управления
- б) информация о состоянии объекта управления
- в) информация о состоянии управляющей системы
- г) команды управления
- д) команды управления и информация об управляющей системе

2. Какой из объектов может являться исполнителем алгоритмов?

- а) карта
- б) словарь
- в) стиральная машина
- г) тетрадь
- д) краска

3. Понятность алгоритма означает, что он должен быть записан с помощью:

- а) команд, понятных создателю алгоритма
- б) команд из системы команд исполнителя
- в) команд, понятных пользователю алгоритма
- г) команд, понятных для компьютера
- д) операторов языка программирования

4. Конечность алгоритма означает, что:

- а) в нем должен присутствовать оператор вывода результата
- б) он должен решать задачу вычислительного характера
- в) в нем должно присутствовать ключевое слово, означающее конец алгоритма
- г) он должен быть применим для решения всех задач заданного типа
- д) результат должен быть получен за конечное число шагов

5. Как называется свойство алгоритма, соответствующее определению: «Алгоритм должен быть записан из команд, понятных исполнителю, каждая команда должна определять однозначное действие исполнителя»?

- а) массовость
- б) точность
- в) конечность
- г) понятность
- д) дискретность

6. Суть такого свойства алгоритма как результативность заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

7. Свойство алгоритма как понятность заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

8. Суть такого свойства алгоритма как *массовость* заключается в том, что:

- а) алгоритм должен иметь дискретную структуру (должен быть разбит на последовательность отдельных шагов);
- б) записывая алгоритм для конкретного исполнителя, можно использовать лишь те команды, что входят в систему его команд;
- в) алгоритм должен обеспечивать решение не одной конкретной задачи, а некоторого класса задач данного типа;
- г) при точном исполнении всех команд алгоритма процесс должен прекратиться за конечное число шагов, приведя к определенному результату;
- д) исполнитель алгоритма не должен принимать решения, не предусмотренные составителем алгоритма.

Итоговая контрольная работа в форме online теста:

Основная ссылка: <https://onlinetestpad.com/ey7lfozwpmijk>

Раздел 2. Основы программирования на языке C#	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
---	--

Тестовое задание «Переменные и типы данных в C#.»

Вариант 1

1. Тип данных определяет...
 - 1) множество значений, которые могут принимать объекты программы
 - 2) множество значений, допустимых для операций программы
 - 3) множество значений, которые могут принимать объекты программы, а также совокупность операций, допустимых над этими значениями
 - 4) совокупность операций, допустимых над этими значениями
2. Символьный тип
 - 1) Char
 - 2) Integer
 - 3) Real
 - 4) Boolean
3. Логический тип имеет значения:
 - 1) да и нет
 - 2) один или несколько
 - 3) истина и ложь
 - 4) истина
4. С помощью какого оператора объявляются переменные контейнера:
 - 1) Dim
 - 2) Private
 - 3) Public
 - 4) Variant
5. Переменная, которая доступна только в текущей процедуре или функции

- 1) локальная
 - 2) глобальная
 - 3) символьная
 - 4) числовая
6. Какая из перечисленных ниже переменных не является числовой
- 1) String
 - 2) Integer
 - 3) Long
 - 4) Single
7. Какое ключевое слово используется при объявлении констант?
- 1) Const
 - 2) Cont
 - 3) Dim
 - 4) Mod
8. Тип переменной Integer обозначает
- 1) логическое значение True или False
 - 2) целые числа от -32768 до 32767
 - 3) целые числа от 0 до 255
 - 4) символы

Вариант 2

1. Что обозначает тип данных Boolean в языке программирования Visual C#?
- 1) этот тип данных может хранить только два значения: True либо False
 - 2) этот тип данных может хранить только два значения: 1 либо 2
 - 3) этот тип данных может хранить любой набор данных
 - 4) такого типа данных не существует
2. Строковый тип
- 1) Хранит строку символов (Верно)
 - 2) Хранит строку чисел
 - 3) Хранит число
 - 4) Хранит выражение
3. В операторе присваивания $summa := \text{sqr}(x) + 3 * a$ переменными являются...
- 1) a, x, summa
 - 2) x, a
 - 3) sqr, x, a
 - 4) summa, sqr, x, a
4. С помощью какого оператора объявляются глобальные переменные:
- 1) Public
 - 2) Private
 - 3) Dim
 - 4) Single
5. Переменные, которые доступны во всем проекте
- 1) переменные контейнера
 - 2) логические
 - 3) локальные
 - 4) глобальные
6. Выберите из перечисленных ниже переменных вещественные:
- 1) String
 - 2) Integer
 - 3) Long
 - 4) Float
7. Резервированное место в оперативной памяти для хранения данных
- 1) оператор
 - 2) переменная
 - 3) свойство
 - 4) метод

8. Величина, не меняющаяся в процессе работы:

- 1) переменная
- 2) константа
- 3) выражение
- 4) процедура

Теоретические вопросы для коллоквиума:

1. Оператор цикла с постусловием Do While.

Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) {s+= 5; i+= 1;}`

2. Оператор цикла с предусловием While.

Определить значения переменных a и b после выполнения следующих операторов: `a= 1; b= 2; while (b<20) {a= a + 2; b= b +2;}`

3. Оператор цикла с параметром For.

Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i< n; i++) s+= 100 / i;`

Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2; i<= n; i++) s+= 100 / i;`

4. Оператор выбора switch.

Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m - n;`

`if (m<=n) p = m + n; else p= 4 - m*n;`

5. Операторы switch и If, назначение и отличие.

Определить значение переменной p после выполнения следующего фрагмента программы: `int m = 13; int n= 21; n= 2*m - n;`

`if (m<=n) p= m + n; else p= 4 - m*n;`

6. Операторы цикла For, While и Do While, блок-схемы.

Какими будут значения переменных s после выполнения следующих операторов: `int s=2; int i= 1; while (i<10) {s+= i; i+= 1;}`

7. Условный оператор IF (IF-....., IF-.....-ELSE).

Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m - n;`

`if (m<=n) p= m + n; else p= 4 - m*n;`

8. Операторы цикла While и Do While, их отличие.

Какими будут значения переменных a и b после выполнения следующих операторов: `int a= 1; int b= 2; while (b<20) { a+= 2; b+= 2;}`

9. Оператор многовероятностного выбора. Форма записи и пример.

Определить значение переменной p после выполнения следующего фрагмента программы: `int m= 13; int n= 21; n= 2*m - n;`

`if (m<=n) p= m + n; else p= 4 - m*n`

10. Ввод одномерных массивов случайным образом. Пример.

Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i< n; i++) s+= 100 / i;`

11. Алгоритм сортировки массива.

Какими будут значения переменных s и i после выполнения следующих операторов: `int s=20; int i= 0; while (i<=10) { s+= 5; i+= 1;}`

12. Ввод двумерных массивов.

Какими будут значения переменных a и b после выполнения следующих операторов:

`int a= 1; int b= 2; while (b<20) { a+= 2; b+=2;}`

13. Вложенные условные операторы. Пример алгоритма.

Определить значение переменной p после выполнения следующего фрагмента программы:

`int m= 13; int n= 21; n= 2*m - n; if (m<=n) p= m + n; else p= 4 - m*n;`

14. Вывод двумерных массивов.

Определить значение переменной s после выполнения следующих операторов: `int s =0; int n =4; for (int i=2; i<= n; i++) s+= 10 / i;`

15. Алгоритм нахождения максимального (минимального) элемента в массиве (одномерный массив, матрица).
16. Алгоритм нахождения порядкового номера максимального (минимального) элемента в массиве (одномерный массив, матрица).
17. Функции деления с остатком. Определить значения переменных a и b после выполнения последовательности действий:
 $a = 4 * 5 / 3 \% 2$; $b = 4 * 5 / (3 \% 2)$;

Практические задания

1. Разработать и произвести отладку программы: Найти сумму бесконечного ряда. Суммировать до тех пор, пока сумма не станет больше заданного $p > 0$. Вывести эти числа.
2. Разработать и произвести отладку программы для определения $N! - M!$. $N! = 1 * 2 * 3 * 4 * \dots * n$
3. Разработать и произвести отладку программы: Вычислить сумму квадратов всех целых чисел, пока сумма квадратов меньше заданного числа A. Вывести эти числа.
4. Разработать и произвести отладку программы: Произведение первых четных чисел равно P, сколько сомножителей взято.
5. Разработать и произвести отладку программы: Определить все двузначные числа, сумма квадратов цифр которых кратны числу 15.
6. Разработать и произвести отладку программы: Даны два одномерных массива одинаковой длины. Получить третий массив такой же размерности, каждый элемент которого равен сумме соответствующих элементов данных массивов.
7. Разработать и произвести отладку программы: ан одномерный массив чисел. Определите сумму элементов, принадлежащих промежутку от A до B (A и B водить с клавиатуры).
8. Разработать и произвести отладку программы определения количества элементов массива, больших среднего арифметического всех его элементов.
9. Разработать и произвести отладку программы: Дан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-10, 10]$. Из элементов массива P сформировать массив M той же размерности по правилу: если номер четный, то $M_i = i * P_i$, если нечетный, то $M_i = -P_i$. Исходный и скорректированный массив вывести на экран.
10. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-30, 30]$. Из элементов массива P сформировать массив M из четных чисел. Исходный и скорректированный массивы вывести на экран.
11. Разработать и произвести отладку программы: ан массив P целых чисел из n элементов, заполненный случайным образом числами из промежутка $[-10, 10]$. Из элементов массива P сформировать массив M той же размерности по возрастанию. Исходный и скорректированный массивы вывести на экран.
12. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке убывания.
13. Разработать и произвести отладку программы, печатающей все делители целого числа в порядке возрастания

Раздел 3 Модульное программирование	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
-------------------------------------	--

Вопросы к коллоквиуму:

1. Файлы (общие сведения, характеристики, описание файловых переменных).
 Определить значение переменной s после выполнения следующих операторов: `int s=0; int n=5; for (int i=2 ; i < n; i++) s+= 100 / i;`
2. Основные процедуры работы с файлами.

- Какими будут значения переменных *a* и *b* после выполнения следующих операторов: `int a=1; int b=2; while (b<20) { a+=2; b+=2;}`
- Процедуры и функции работы с текстовыми файлами.
Какими будут значения переменных *a* и *b* после выполнения следующих операторов: `int a=1; int b=1; while (a<=3) { a+=1; b+=1;}`
 - Основные команды работы с графикой.
Чему равны значения переменных *a* и *b* после выполнения последовательности действий: `int a=15/(16%7); int b=34%a*5-29%5*2;`
 - Текстовые файлы (назначение, описание в программе).
Какими будут значения переменных *s* и *i* после выполнения следующих операторов: `int s=20; int i=0; while (i<=10) { s+=5; i+=1;}`

Практические задания для самостоятельной работы:

- Разработать и произвести отладку программы: Пользователь угадывает число задуманное компьютером, при помощи подсказок больше или меньше, компьютер выдаёт количество шагов, за которые пользователь угадал число.
- Разработать и произвести отладку программы: Произведение *N* первых нечетных чисел равно *p*. Сколько сомножителей взято?
- Дана последовательность из *N* целых чисел. Определить произведение максимального и минимального элементов этой последовательности.
- Составьте программу- "перевертыш" (так называются слова, читающиеся одинаково слева направо и справа налево, например: ПОТОП, КАЗАК).
- Дан текст со скобками, удалить текст в скобках вместе со скобками и вывести его отдельно.
- Вывести все двухзначные числа, сумма цифр которых равна *N*.
- В тексте имеется одна точка с запятой «;». Подсчитать количество символов до точки с запятой и после нее.
- В тексте имеются запятые. Подсчитать количество запятых.
- Вывести все четырехзначные числа, у которых сумма первых двух равна сумме двух последних.
- В тексте имеются «;». Подсчитать их количество.
- Вывести все шестизначные числа, у которых сумма первых трех равна сумме трех последних.
- В тексте имеются скобки. Подсчитать количество скобок открывающихся и закрывающихся.
- Вывести все четырехзначные числа, сумма цифр которых равна *N*.
- Поменять порядок следования цифр в натуральном числе *N* на обратный.

Раздел 4. Объектно-ориентированное программирование	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
---	--

Теоретические вопросы для коллоквиума:

- Интегрированная среда разработки Visual Studio.
- Принципы объектно-ориентированного программирования: инкапсуляция.
- Принципы объектно-ориентированного программирования: Понятие класса и объекта.
- Назначение и свойства компонента CheckBox.
- Организация проекта. Структура проекта: файл проекта, файлы настроек проекта.
- Обработка исключительных ситуаций. Классификация
- Принципы объектно-ориентированного программирования: наследование.
- Организация проекта. Интерфейс приложения: форма как главный элемент приложения, основные свойства формы.

9. Структура компонентов: Свойства. Методы.
10. Компиляция и отладка программы: Типы сообщений компилятора. Использование встроенного отладчика. Трассировка программы.
11. *Управление проектом: создание, сохранение, открытие и настройка проектов.
12. Принципы объектно-ориентированного программирования: полиморфизм.
13. Принципы объектно-ориентированного программирования: Методы. Вызов методов.
14. Свойства и обработчик событий.
15. Принципы объектно-ориентированного программирования: Свойства. Описание свойства.
16. Организация проекта. Управление проектом: создание, сохранение и открытие, настройки.
17. Организация проекта. Компиляция и отладка программы: типы сообщений компилятора, использование встроенного отладчика.
18. Управляющие элементы: списки, кнопки, меню, переключатели выбора, таблицы, панель инструментов.
19. Интерфейс окна Visual Studio.
20. Интерфейс приложения: Форма как главный элемент приложения. Основные свойства формы. Размещение компонентов на форме.
21. Структура проекта: файл ресурсов, файла исходного модуля и формы, файлы создаваемые пользователем.
22. Структура проекта: Файл проекта. Файлы настроек проекта. Файл ресурсов.
23. Назначение команды меню File.
24. Организация проекта. Управление проектом: создание, сохранение и открытие, настройки.
25. Дизайнер форм. Редактор кода.
26. Назначение компонента Timer.
27. Назначение и свойства компонентов TextBox и Label.
28. Основные принципы ООП.
29. Назначение вкладки Standard палитры компонентов.
30. Свойства. Методы. События и обработчики событий.
31. Разработка интерфейса приложения, обработка событий.
32. Назначение компонента RadioButton.
33. Назначение компонента ListBox.
34. Компоненты вывода данных.
35. Назначение и свойства компонента Button.

Тест: «Основные элементы Visual V# и принципы программирования»

Основная ссылка: <https://onlinetestpad.com/lhgiiiumi4cgs>

Промежуточная аттестация в форме контрольной работы – 2 семестр	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
---	--

Задания для контрольной работы:

1. Разработать и произвести отладку программы для решения квадратного уравнения.
2. Создать и отладить приложение – конвертор перевода суммы денег из долларов в рубли.
3. Разработать и произвести отладку программы для вычисления делителей натурального числа N. Вывести сами делители, их количество.
4. Разработать и произвести отладку программы, вычисляющей сумму 1-й и последней цифр натурального числа N. Вывести эти цифры и сумму.
5. Создать и отладить приложение для решения квадратного уравнения.
6. Разработать и произвести отладку программы, находящей все простые числа в заданном диапазоне.

7. Разработать и произвести отладку программы, находящей все нечетные числа в заданном диапазоне и их количество.
8. Разработать и произвести отладку программы, находящей все четные числа в заданном диапазоне и их количество.
9. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; заменить отрицательные числа на 0, положительные – на 1.
10. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по убыванию.
11. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; отсортировать массив по возрастанию.
12. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива; поменять местами два элемента массива с номерами k1 и k2.
13. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет минимальный и максимальный элементы массива.
14. Разработать и произвести отладку программы, которая задает размер линейного массива, заполняет этот массив случайными целыми числами, выводит список элементов массива, определяет сумму всех элементов и количество положительных элементов.

Промежуточная аттестация в форме комплексного дифференцированного зачета– 3 семестр	ПК 1.1.; ОК 01.; ПК 1.2.; ОК 02.; ПК 1.4.; ОК 04.; ОК 05.; ПК 1.5.; ОК 09.; ПК 2.4.; ПК 2.5.; ПК 11.1.; ПК 11.2.
---	--

Вопросы к комплексному дифференцированному зачету

1. Понятие алгоритма и алгоритмизации. Основные свойства алгоритмов.
2. Формы представления алгоритмов. Примеры
3. Понятие блок-схемы, основные блочные символы.
4. Псевдокод и алгоритмический язык. Исполнители алгоритмов.
5. Что такое событие? Пример.
6. Что такое метод? Пример.
7. Встроенные типы данных Visual C#.
8. Методы и свойства класса Console
9. Создание проекта Windows forms C# в среде Microsoft Visual Studio
10. Основные принципы ООП: полиморфизм, инкапсуляция, наследование.
11. Что такое локальная и глобальная переменные. Как объявляется локальная переменная?
12. Перечислите основные типы переменных. Используя какие, операторы можно объявить переменные?
13. Назовите основные события, связанные с клавиатурой и мышью.
14. Встроенные функции в Visual C#.
15. Назовите основные арифметические операции языка Visual C#.
16. Общий синтаксис объявления процедуры.
17. Операторные скобки {}.
18. Класс Directory. Основные методы класса Directory.

19. Основные методы класса Path.
20. Графика в программировании. Объект Pen. Кисти Brush. Методы рисования графических примитивов.
21. Дать определения: объектно-ориентированное программирование, класс, объект.
22. Дать определения: инкапсуляция, наследование, полиморфизм.
23. Дать определения: свойство, метод, событие, процедура.
24. Дать определения: оператор, переменная.
25. Что такое среда разработки. Что она содержит?
26. Целочисленные типы данных. Вещественные типы данных. Десятичный тип данных.
27. Логический тип данных. Символьный тип данных. Строковый тип данных.
28. Объявление и инициализация переменной в C#. Неявно типизированные переменные. Константы.
29. Операции и выражения в C#. На какие виды делятся операции по количеству участвующих операндов. Привести несколько примеров.
30. Для чего предназначен класс Math. Пример обращения к членам класса Math. Привести несколько примеров методов класса Math.
31. Условный оператор If.
32. Оператор выбора switch и оператор прерывания break.
33. Циклы в C#. Оператор for. Структура цикла, организованного с помощью этого оператора.
34. Оператор цикла while. Оператор цикла do...while.
35. Операторы прерываний break (для циклов) и continue
36. Диалоговые окна. Типы диалоговых окон. Класс MessageBox.
37. Отладка программ. Какие ошибки могут возникать в программе? Трассировка. Точка останова.
38. Обработка исключений. Примеры типов исключений.
39. Массив. Виды массивов. Объявление и инициализация массива. Многомерные массивы.
40. Класс System.Array. Основные элементы класса Array.
41. Работа с символами в C#. Основные методы класса System.Char.
42. Обработка строк текста в C#.
43. Графика в программировании. Методы рисования графических примитивов.
44. Класс File. Некоторые члены класса File.
45. Работа с информацией о файле. Работа с каталогами и устройствами. Работа с путями к файлам.