

Приложение ППСЗ/ППКРС по специальности 09.02.07 Информационные системы и
программирование (специалист по информационным системам)
2022-2023 уч.г.: Комплект контрольно-оценочных средств учебной дисциплины ОП.04 Основы
алгоритмизации и программирования

МИНИСТЕРСТВО ОБРАЗОВАНИЯ БЕЛГОРОДСКОЙ ОБЛАСТИ
ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«АЛЕКСЕЕВСКИЙ КОЛЛЕДЖ»

**Комплект
контрольно-оценочных средств**

по учебной дисциплине

ОП.04 Основы алгоритмизации и программирования

для специальности

**09.02.07 Информационные системы и программирование
(специалист по информационным системам)**

Комплект контрольно-оценочных средств разработан на основе Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.07 Информационные системы и программирование

Составитель:

Жук Наталья Михайловна, преподаватель

1. Паспорт комплекта оценочных средств

1.1 Область применения комплекта оценочных средств

Контрольно-оценочные средства (КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины ОП.04 Основы алгоритмизации и программирования.

КОС включают контрольные материалы для проведения промежуточной аттестации в форме дифференцированного зачета.

КОС разработан на основании рабочей программы учебной дисциплины ОП.04 Основы алгоритмизации и программирования.

1.2. Цели и задачи учебной дисциплины – требования к результатам освоения учебной дисциплины:

В результате освоения учебной дисциплины обучающийся должен **уметь:**

1. разрабатывать алгоритмы для конкретных задач;
2. использовать программы для графического отображения алгоритмов;
3. определять сложность работы алгоритмов;
4. работать в среде программирования;
5. реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
6. оформлять код программы в соответствии со стандартом кодирования;
7. выполнять проверку, отладку кода программы.

В соответствии с ФГОС СПО в результате освоения учебной дисциплины обучающийся должен **знать:**

1. понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
2. эволюцию языков программирования, их классификацию, понятие системы программирования;
3. основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
4. подпрограммы, составление библиотек подпрограмм;
5. 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 переопределения.

Профессиональные (ПК) и общие (ОК) **компетенции**, которые актуализируются при изучении учебной дисциплины:

ОК.1 Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам

ОК.2 Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК.4 Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК.5 Пользоваться профессиональной документацией на государственном и иностранном языках

ОК.9 Использовать информационные технологии в профессиональной деятельности.

ОК.10 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 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 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.

Перечень знаний, умений, навыков в соответствии со спецификацией стандарта компетенции Ворлдскиллс Программные решения для бизнеса, которые актуализируются при изучении учебной дисциплины:

1) знать необходимость быть в курсе новых технологий и принимать решение о целесообразности их применения; важность оптимизации архитектуры системы с учетом модульности и повторного использования;

2) уметь анализировать системы с помощью структурного моделирования и анализа (например, объекты, классы, диаграммы классов предметной области).

1.4. Планируемые личностные результаты освоения рабочей программы

ЛР 2. Проявляющий активную гражданскую позицию, демонстрирующий приверженность принципам честности, порядочности, открытости, экономически активный и участвующий в студенческом и территориальном самоуправлении, в том числе на условиях добровольчества, продуктивно взаимодействующий и участвующий в деятельности общественных организаций.

ЛР 3. Соблюдающий нормы правопорядка, следующий идеалам гражданского общества, обеспечения безопасности, прав и свобод граждан России. Лояльный к установкам и проявлениям представителей субкультур, отличающий их от групп с деструктивным и девиантным поведением. Демонстрирующий неприятие и предупреждающий социально опасное поведение окружающих.

ЛР 4. Проявляющий и демонстрирующий уважение к людям труда, осознающий ценность собственного труда. Стремящийся к формированию в

сетевой среде лично и профессионального конструктивного «цифрового следа».

1.3 Результаты освоения учебной дисциплины, подлежащие проверке

| Наименование разделов\тем | Коды умений (У), знаний (З), личностных результатов (ЛР), формированию которых способствует элемент программы | Средства контроля и оценки результатов обучения в рамках текущей аттестации (номер задания) | Средства контроля и оценки результатов обучения в рамках промежуточной аттестации (номер задания/контрольного вопроса/ экзаменационного билета) |
|--|---|---|---|
| Раздел 1. Введение в программирование | У1 З1 З2 З3 ЛР 2 ЛР 3 ЛР 4 | Выполнение заданий | Выполнение заданий по темам: Тема 1.1. Языки программирования Тема 1.2. Типы данных |
| Раздел 2. Основные элементы кода программы - операторы языка программирования | У1 У2 У3 З2 З3 ЛР 2 ЛР 3 ЛР 4 | Выполнение заданий | Выполнение заданий по темам: Тема 2.1. Операторы языка программирования |
| Раздел 3. Современные стили программирования. | У2-5 З1 З2 З3 ЛР 2 ЛР 3 ЛР 4 | Выполнение заданий | Выполнение заданий по темам: Тема 3.1. Процедуры и функции Тема 3.2. Структуризация в программировании Тема 3.3. Модульное программирование Тема 3.4. Указатели |
| Раздел 4. Разработка объектно-ориентированных программ. | У2-6 З1 З2 З3-5 ЛР 2 ЛР 3 | Выполнение заданий | Выполнение заданий по темам: Тема 4.1. Основные принципы объектно-ориентированного программирования |

| | | | |
|--|---|--------------------|---|
| | ЛР 4 | | (ООП). Тема 4.2. Интегрированная среда разработчика Тема 4.3. Визуальное событийно-управляемое программирование Тема 4.4. Разработка оконного приложения Тема 4.5. Этапы разработки приложений Тема 4.6. Иерархия классов |
| Раздел 5. Практическая работа по программированию | У 4-7 31 32 34-5 ЛР 2 ЛР 3 ЛР 4 | Выполнение заданий | Выполнение заданий по темам: Тема 5.1. Основы работы с процедурами и функциями Тема 5.2. Практика работы с вызовом функций и процедур, вложенных функций и рекурсивных алгоритмов Тема 5.3. Работы со стандартными модулями и с модулями обработки простых типов данных Тема 5.4. Работа с модулями обработки структурированных данных и пользовательскими модулями Тема 5.5. Работа с указателями Тема 5.6. Работа с интегрированной средой разработчика Тема 5.7. Работа с |

| | | | |
|--|--|--|--|
| | | | визуальным событийно- управляемым программированием Тема 5.8. Разработка многооконного приложения Тема 5.9. Практика разработки оконного приложения Тема 5.10. Выполнение этапов разработки приложений Тема 5.11. Выполнение этапов разработки приложений |
|--|--|--|--|

2. Комплект оценочных средств для текущей аттестации

2.1. Примерные практические задания (ПЗ)

1. Необходимо написать консольную программу, где пользователь будет вводить с клавиатуры 2 числа. Числа будут сравниваться с последующим выводом в консоль результата этого сравнения (равны ли значения, а если нет, то какое число больше/меньше).

2. Пусть пользователь имеет возможность ввести одно число, причем:
 — когда значение больше пяти и одновременно с этим меньше десяти, выводится один текст с соответствующим сообщением;

— когда значение не попадает в указанный диапазон, программа его не узнает, выводя другой текст с соответствующим сообщением.

3. Представим банк, в котором алгоритм начисления процента по вкладу зависит от суммы вклада. На вход будет подаваться число (сумма вклада). При значении меньше 100, будет начислено 5 %, если значение находится в диапазоне от ста до двухсот — 7 %, если больше — 10 %. Отработав, программа должна вывести общую сумму с начисленными процентами. Для решения этой задачи воспользуемся выражением `Convert.ToDouble(Console.ReadLine())`, которое нам пригодится для получения вводимого с клавиатуры числа.

4. Пусть банк регулярно начисляет по всем вкладам не только положенные по договору проценты, но еще и бонусы. И пусть, к примеру, банк решит выполнить доначисление по всем клиентским вкладам в размере 15 единиц без учета суммы. Поменяем программу, отображенную выше, чтобы к итоговой сумме были добавлены еще и бонусы.

5. Дана сторона квадрата a . Найти его периметр $P = 4 \cdot a$.

6. Дана сторона квадрата a . Найти его площадь $S = a^2$.

7. Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.

8. Дан диаметр окружности d . Найти ее длину $L = \pi \cdot d$. В качестве значения π использовать 3.14.

9. Дана длина ребра куба a . Найти объем куба $V = a^3$ и площадь его поверхности $S = 6 \cdot a^2$.

10. Даны длины ребер a , b , c прямоугольного параллелепипеда. Найти его объем $V = a \cdot b \cdot c$ и площадь поверхности $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$.

11. Найти длину окружности L и площадь круга S заданного радиуса R : $L = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$. В качестве значения π использовать 3.14.

12. Даны два числа a и b . Найти их среднее арифметическое: $(a + b)/2$.

13. Даны два неотрицательных числа a и b . Найти их среднее геометрическое, то есть квадратный корень из их произведения: $\sqrt{a \cdot b}$.

14. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.

15. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их модулей.

16. Даны катеты прямоугольного треугольника a и b . Найти его гипотенузу c и периметр P : $c = \sqrt{a^2 + b^2}$, $P = a + b + c$.

17. Даны два круга с общим центром и радиусами R_1 и R_2 ($R_1 > R_2$). Найти площади этих кругов S_1 и S_2 , а также площадь S_3 кольца, внешний радиус которого равен R_1 , а внутренний радиус равен R_2 : $S_1 = \pi \cdot (R_1)^2$, $S_2 = \pi \cdot (R_2)^2$, $S_3 = S_1 - S_2$. В качестве значения π использовать 3.14.

18. Дана длина L окружности. Найти ее радиус R и площадь S круга, ограниченного этой окружностью, учитывая, что $L = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$. В качестве значения π использовать 3.14.

19. Даны координаты двух противоположных вершин прямоугольника: (x_1, y_1) , (x_2, y_2) . Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.

20. Найти расстояние между двумя точками с заданными координатами (x_1, y_1) и (x_2, y_2) на плоскости. Расстояние вычисляется по формуле $q = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$.

21. Даны координаты трех вершин треугольника: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь, используя формулу для расстояния между двумя точками на плоскости. Для нахождения площади треугольника со сторонами a , b , c использовать формулу Герона: $S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}$, где $p = (a + b + c)/2$ — полупериметр.

22. Поменять местами содержимое переменных A и B и вывести новые значения A и B .

23. Даны переменные A , B , C . Изменить их значения, переместив содержимое A в B , B — в C , C — в A , и вывести новые значения переменных A , B , C .

24. Даны переменные A , B , C . Изменить их значения, переместив содержимое A в C , C — в B , B — в A , и вывести новые значения переменных A , B , C .

25. Найти значение функции $y = 3x^6 - 6x^2 - 7$ при данном значении x .

26. Найти значение функции $y = 4(x-3)^6 - 7(x-3)^3 + 2$ при данном значении x .

27. Дано число A . Вычислить A^8 , используя вспомогательную переменную и три операции умножения. Для этого последовательно находить A^2 , A^4 , A^8 . Вывести все найденные степени числа A .

1. Дополнить класс `Fraction` перегруженными арифметическими операциями, в которых один из операндов является вещественным числом.

2. Дополнить класс `Fraction` перегруженными операциями сравнения для дробей и вещественных чисел.

3. Дополнить класс `Fraction` перегруженным конструктором, осуществляющим преобразование вещественного числа к типу `Fraction`. Предполагается, что дробная часть вещественного числа содержит до 10 знаков после запятой.

4. Разработать класс «Комплексное число». Определить в нем конструктор, перегрузить арифметические операции, операции сравнения, операцию преобразования в строку и статический метод получения комплексного числа из строки.

5. Разработать класс «Комплексное число в тригонометрической форме». Определить в нем конструктор, перегрузить арифметические операции, операции сравнения, операцию преобразования в строку и статический метод получения комплексного числа из строки.

6. Разработать класс «Комплексное число», в котором данные хранятся в двух видах: алгебраической и тригонометрической формах. Определить в нем конструкторы и деструктор, перегрузить арифметические операции, операции сравнения, операцию преобразования в строку и статический метод получения комплексного числа из строки, написать методы преобразования числа из одной формы в другую. Протестировать все возможности класса.

7. Разработать класс «Дата». Определить в нем конструкторы и деструктор, перегрузить операцию добавления к дате заданного количества дней, операцию вычитания двух дат, операции сравнения и операцию преобразования в символьную строку, а также статический метод получения даты из строки.

8. Разработать класс «Время». Определить в нем конструкторы и деструктор, перегрузить операцию добавления к времени заданного количества минут, операцию вычитания двух моментов времени, операцию преобразования в символьную строку и метод получения момента времени из строки.

9. Разработать класс «Прямоугольник». Определить в нем конструкторы и деструктор, перегрузить операцию пересечения прямоугольников (операция «*»), операцию вычисления площади прямоугольника операции сравнения (по площади), операцию преобразования в символьную строку и метод получения объекта прямоугольника из строки.

10. Разработать класс «Треугольник». Определить в нем конструкторы и деструктор, перегрузить операцию преобразования в вещественное число (площадь треугольника), операцию проверки включения точки в треугольник, операции сравнения треугольников (по площади), операцию преобразования в символьную строку и метод получения объекта-треугольника из строки.

11. Создать класс «Студент», который определяется полями ФИО, номер группы, название факультета, название специальности, средний балл успеваемости. Пусть имеется массив объектов этого класса. Разработать метод выбора студентов из массива по условию (учится на конкретном факультете, имеет средний балл более заданного уровня и пр.). Для определения, удовлетворяет ли объект условию, передать в метод параметр-делегат.

12. Для массива объектов класса «Студент» из задания 1 создать метод сортировки по различным критериям (по фамилии, по среднему баллу успеваемости). Метод сравнения двух объектов передать в метод сортировки как параметр-делегат.

13. Пусть имеется класс «Матрица». Определить различные методы, которые осуществляют преобразование матрицы (транспонирование, поворот, сортировка строк, изменение порядка столбцов на обратный и пр.). В диалоговом режиме задать последовательность действий, которую нужно произвести с объектом матрицей, сохраняя ее в переменной-делегате. Предусмотреть команду меню выполнения этих действий, которая обращается к сформированному делегату.

14. Разработать класс для решения задачи поиска точки, минимизирующей функцию на отрезке. Реализовать функции решения этой задачи различными методами – методом деления отрезка пополам, методом золотого сечения, методом касательных, методом Фибоначчи. Выбранный способ решения сохранить в делегат.

15. Разработать класс «Множество». Определить конструкторы и деструктор. Переопределить операции объединения, пересечения и разности двух множеств, методы для организации ввода-вывода. Написать методы проверки включения одного множества в другое, проверки равенства двух множеств, проверки пустоты множества.

16. Разработать класс «Многоугольник», который хранится в виде массива его вершин. Определить конструктор, методы для организации ввода-вывода и переопределить операции сравнения многоугольников по площади. Написать методы вычисления площади многоугольника, определения, принадлежит ли точка многоугольнику, определения, является ли многоугольник выпуклым.

17. Разработать класс «Целое число в заданной системе счисления». Число должно храниться в виде массива целых чисел (разрядов числа). Определить конструктор, методы для организации ввода-вывода, операции сложения, вычитания, умножения, деления и взятия остатка от деления двух чисел и операции сравнения. Написать методы перевода числа из одной системы счисления в другую.

18. Разработать класс «Бинарное дерево сортировки». Написать

конструкторы и деструктор, методы добавления нового узла, удаления узла по ключевому значению, вычисления глубины дерева, объединения двух деревьев, вычисления количества узлов на заданном уровне, определения подобия двух деревьев.

19. Раскрыть интерфейс `IMathObject` на примере класса «Рациональное число». Протестировать обращение к наследуемым методам без явного указания на тип объектов.

20. Раскрыть интерфейс `IMathObject` на примере класса «Комплексное число». Протестировать обращение к наследуемым методам без явного указания на тип объектов.

21. Для массива объектов, которые раскрывают интерфейс `IMathObject`, создать метод-обобщение, который находит сумму объектов массива. Протестировать метод для массива матриц, массива комплексных чисел и массива полиномов.

22. Разработать интерфейс «Фигура на плоскости». Определить для него операции перемещения, поворота, определения площади, получения местоположения и пр. Раскрыть интерфейс в классах «Треугольник», «Прямоугольник», «Многоугольник». 5. Разработать класс «Линейная функция в n-мерном пространстве» ($f(x) = b, x + c$). Определить конструктор, переопределить операции сложения и вычитания функций, умножения функции на число. Для организации ввода-вывода переопределить операцию преобразования в строку и статический метод `Parse()`. Написать методы вычисления значения функции в точке, получения градиента функции. Наследовать от этого класса класс «Квадратичная функция в n-мерном пространстве» ($f(x) = Ax, x + b, x + c$). Переопределить все указанные операции и методы для класса наследника. 6. Разработать класс «Граф» в виде списка смежности. Определить конструкторы и деструктор. Переопределить операции ввода-вывода. Написать методы проверки связности графа, проверки полноты графа, проверки двудольности графа, получения дополнения графа, нахождения источника графа, нахождения стока графа. Наследовать от этого класса класс «Взвешенный граф». Написать методы получения кратчайшего пути между двумя вершинами алгоритмом Дейкстры, получения каркаса минимального веса алгоритмами Прима и Краскала.

2.2. Тестовые задания (ТЗ)

1. Можно ли сделать перегрузку операторов `true` и `false`?

- Да
 Нет

2. Что будет выведено на консоль?

```
1    public class Person
2    {
3        public readonly string name = "Artur";
4    }
```

```

5     public Person(string name)
6     {
7         this.name = name;
8     }
9 }
10
11 class Program
12 {
13     static void Main(string[] args)
14     {
15         Person person = new Person("John");
16         Console.WriteLine(person.name);
17
18         Console.ReadKey();
19     }
20 }

```

- "John"
- "Artur"
- Возникнет ошибка

3. Можно ли наследоваться от нескольких классов?

- Да
- Нет

4. Возникнет ли ошибка при выполнении данного фрагмента кода?

```

1     object model = 156;
2     model = "157";
3     Console.WriteLine(model);

```

- Нет
- Да

5. C# код компилируется в код ...

- CIL
- MASM
- CLI
- FCL

6. Выполнится ли блок finally, если исключения не было?

- Зависит от режима работы
- Нет
- Зависит от кода
- Да

7. Можно ли сделать перегрузку оператора присваивания?

- Нет
- Да

8. Что будет выведено на консоль?

```
1 int firstNumber = 5;
2 double secondNumber = 5.0;
3 Console.WriteLine(firstNumber == secondNumber);
```

- False
- Возникнет ошибка
- True

9. Что будет выведено на консоль?

```
1 int number = 2;
2 Console.WriteLine(++number - number-- * ++number);
```

- 6
- 1
- 0
- 1
- Возникнет ошибка

10. Что будет выведено на консоль?

```
1 bool isValid = 1;
2 Console.WriteLine(isValid);
```

- 1
- false
- true
- Возникнет ошибка

11. Что будет выведено на консоль?

```
1 float digit = 0 / 0f;
2 Console.WriteLine(digit == digit);
```

- Возникнет ошибка
- False
- NaN
- True

12. Что делает модификатор partial?

- Запрещает наследование
- Позволяет разделять сущность на несколько частей
- Позволяет использовать неуправляемый код
- Ограничивает видимость текущим классом

13. Какой способ вызова исключения является корректным?

- throw Exception();
- call Exception();
- throw new Exception();

call new Exception();

new Exception();

Exception();

15. Что будет выведено на консоль?

```
1 Console.WriteLine(Int16.MaxValue);
```

32767

2147483647

255

9223372036854775807

16. Что будет выведено на консоль?

```
1 Console.WriteLine("Text" + " test");
```

Возникнет ошибка

String.Empty

"Text test"

"Text"

" test"

17. Что будет содержать строка name после выполнения данного кода?

```
1 string name = "Alexey";
```

```
2 name[1] = ".";
```

"A.alexey"

Возникнет ошибка

"Alexey"

".alexey"

18. Можно ли наследоваться от нескольких интерфейсов?

Нет

Да

Верно ли, что тип string хранит набор символов Unicode?

Да

Нет

19. Что будет выведено на консоль?

```
1 public static void Main(string[] args)
```

```
2 {
```

```
3 {
```

```
4     int digit = 5;
```

```
5 }
```

```
6
```

```
7     Console.WriteLine(digit);
```

```
8 }
```

5

Возникнет ошибка

0

""

null

20. Что будет выведено на консоль?

```
1 byte firstValue = 200;  
2 byte secondValue = 100;  
3 Console.WriteLine(firstValue + secondValue);
```

-45

44

Возникнет ошибка

300

-44

45

21. Что будет выведено на консоль?

```
1 Console.WriteLine(11 / 2);
```

5.5

5

6

22. Что будет выведено на консоль?

```
1 int number = -3;  
2 Console.WriteLine((uint)number);
```

4294967293

Возникнет ошибка

3

-3

0

23. Что будет выведено на консоль?

```
1 dynamic number = 3002;  
2 Console.WriteLine(number.GetType().Name);
```

Dynamic

Single

Int

Object

Int32

24. Структура — это ...

Ссылочный тип

Значимый тип
25. Тип string является значимым или ссылочным типом?

Значимым

Ссылочным

26. Что делает модификатор sealed?

Добавляется просто для наглядности, но ничего не изменяет

Разрешает наследование только один раз

Запрещает наследование

Ограничивает видимость текущей сборкой

27. Что будет выведено на консоль?

```
1  enum Names
2  {
3      Viktor
4  }
5
6  static void Main(string[] args)
7  {
8      Console.WriteLine(Names.Viktor.GetType().Name);
9  }
```

Enum

Value

Object

EnumValue

Names

28. Что будет находиться в переменной name?

```
1  int? a = null;
2  string name = a == null ? "Artur" : "Vasya";
```

Возникнет ошибка

"Artur"

String.Empty

null

"Vasya"

29. Что будет выведено на консоль?

```
1  class Person
2  {
3      public string Name { get; set; }
4
5      public Person(string name)
6      {
7          this.Name = name;
8      }
9  }
```



```

10
11 class Program
12 {
13     static void Main(string[] args)
14     {
15         const string personName = "Anatoliy";
16
17         Person firstPerson = new Person(personName);
18         Person secondPerson = new Person(personName);
19
20         Console.WriteLine(firstPerson == secondPerson);
21     }
22 }

```

- False
- True
- Возникнет ошибка

30. Что будет содержать массив models после выполнения данного фрагмента кода?

```

1 object[] models = { 1, "300", 100, "5" };
2 Array.Sort(models);

```

- Возникнет ошибка
- 1, "5", 100, "300"
- 1, 100, "300", "5"
- 1, "300", 100, "5"

3. Комплект оценочных средств для промежуточной аттестации

3.1. Практические задания (ПЗ)

1. Подсчитать сумму элементов над главной диагональю квадратной матрицы и под ней.
2. Определить максимальный и минимальный элементы двумерного массива и их индексы.
3. Заполнить одномерный массив случайными неповторяющимися числами.
4. Написать программу, реализующую игру «Угадай число».
5. Написать программу решения квадратного уравнения.
6. Расположить строки двумерного массива в порядке возрастания сумм элементов в них.
7. Заполнить одномерный массив случайными числами из интервала $[-10, 10]$. Упорядочить массив по возрастанию, если сумма элементов массива положительна, или по убыванию в противном случае.

8. Определить сумму n первых членов последовательности: $\frac{1}{2}, \frac{2}{3}, \dots, \frac{n}{n+1}$.
9. Сохранить данные одномерного массива в файле данных. Организовать чтение данных из файла.
10. Преобразовать введенное число в строку (123 \rightarrow «123»).
11. Решить линейное неравенство, не используя условную конструкцию.
12. Определить количество слов во введенном тексте.
13. Заполнить двумерный массив случайными числами. Вывести массив на экран. Найти максимальный и минимальный элементы, их индексы.
14. Заполнить одномерный массив случайными числами. Упорядочить массив по возрастанию или убыванию по желанию пользователя.
15. Подсчитать, сколько среди пятизначных чисел таких, у которых сумма четных цифр равна сумме нечетных цифр.
16. Определить, сколько раз в тексте встречается указанный фрагмент.
17. Создать проект, позволяющий сохранять произвольный текст в текстовом файле с указанным пользователем именем. Загружать текст из файла.
18. Создать проект, позволяющий сохранять текст, набранный в текстовом окне в файле данных, очищать текстовое окно и загружать в него сохраненный текст.
19. Создать проект, позволяющий построить треугольник по координатам вершин.
20. Имеется десять флажков. Создать проект, позволяющий сохранять состояние флажков, обнулять и восстанавливать сохраненное состояние.
21. Организовать анимацию процесса движения шарика по окружности.
22. Создать проект, реализующий часы с «будильником».
23. Проект содержит три текстовых окна. Используя буфер обмена, организовать обмен информацией между приложениями.
24. Вывести алфавит введенного пользователем текста.
25. Создать проект, позволяющий добавлять записи в простой список и удалять их.
26. Расположить слова строкового массива в порядке возрастания их длин.
27. Построить на экране правильный N -угольник. Количество вершин и радиус окружности, на которой они расположены, указываются пользователем.
28. Построить график функции в выбранной пользователем системе координат.
29. Составить программу, генерирующую цвет по трем его составляющим (красный, зеленый, синий).
30. Составить программу, моделирующую работу светофора.

31. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран в любую его точку правильный многоугольник произвольного размера с произвольным количеством вершин.

32. Написать программу, моделирующую движение броуновской частицы.

33. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран график функции. График должен выводиться в прямоугольную область указанного размера. Объект должен иметь свойство, определяющее величину единичного отрезка.

34. Разработать класс, объекты которого фиксируют время своего существования.

35. Разработать класс, объекты которого обладают методами, позволяющими вывести на экран анимацию вращающегося шарика. Координата точки – оси вращения, скорость вращения, радиус вращения и направление вращения задаются.

36. Описать процедуру, имеющую три аргумента целого типа и работающую по следующему принципу: значения трех переменных целого типа, описанных выше точки вызова и указанных в качестве аргументов процедуры, должны быть расположены по возрастанию в порядке указания аргументов в процедуре.

37. Разработать класс, объекты которого обладают методами, позволяющими заполнить одномерный массив случайными неповторяющимися числами, вывести его на экран и упорядочить.

38. Решить квадратное уравнение, используя директивы препроцессора и условную операцию.

39. Описать процедуру, позволяющую произвести обмен значениями переменных указанных в качестве аргумента.

40. Разработать компонент, предлагающий интерфейс для отображения одномерного массива и имеющий методы для заполнения, очистки и сортировки.

41. Разработать компонент «Переключатель», имеющий пять стационарных состояний. Компонент должен иметь свойство для установки состояния и обрабатывать событие, связанное с выбором другого состояния.

42. На форме имеется десять командных кнопок (пронумерованных от 1 до 10) и графический контейнер. При нажатии на одну из кнопок в контейнере должно появиться соответствующее римское число.

43. Написать программу, позволяющую просматривать записи простейшей базы данных, добавлять, корректировать и удалять их.

44. Разработать программу, позволяющую заполнять двумерный массив с указанным пользователем количеством строк и столбцов и находить среднее арифметическое в каждой строке и в каждом столбце.

45. Используя общий диалог, написать программу, позволяющую сохранять указанный пользователем текст в файле и считывать сохраненный текст из файла.

46. Используя объектно-ориентированный подход, написать программу, моделирующую движение броуновских частиц.

47. Разработать компонент «Таймер», выводящий время, прошедшее с момента запуска таймера. Компонент должен иметь метод, возвращающий текущее время, обрабатывать событие, связанное с включением и выключением таймера и метод, обнуляющий таймер.

48. Описать класс, объекты которого связываются с текстовыми окнами, хранят введенную пользователем в окно информацию, обладают методом сортировки символов в текстовых окнах в алфавитном порядке и методом восстанавливающим текст в окне после сортировки.

49. Разработать компонент, предлагающий интерфейс для ввода данных и вывода решения квадратного уравнения.

50. Разработать компонент, представляющий собой окно, в котором можно рисовать окружности. Компонент должен иметь свойство, определяющее цвет «графического пера».

51. Разработать проект, моделирующий арифметический калькулятор. Калькулятор должен выполнять следующие операции: сложение, вычитание, умножение, деление, возведение в квадрат и извлечение квадратного корня как для целых, так и для вещественных чисел.

52. Создать проект, моделирующий работу секундомера. Секундомер должен иметь циферблат с минутной и секундной стрелками, метку и одну командную кнопку.

53. Разработайте проект, имитирующий работу простейшего графического редактора.

3.2. Контрольные вопросы (КВ)

1. Принципы структурного программирования. Блочный (модульный) подход к построению алгоритмов решения задач.

2. Процедуры. Вызов процедуры в программах. Формальные и фактические параметры. Соответствие между параметрами. Примеры.

3. Параметры-переменные, параметры-значения, параметры-константы. Примеры.

4. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры.

5. Область действия переменных и других идентификаторов.

6. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.

7. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.

8. Функции, ориентированные на работу со строками. Примеры.

9. Процедуры, ориентированные на работу со строками. Примеры.

10. "Пузырьковая" сортировка (обменами).

11. Сортировка выбором.

12. Сортировка вставками.

13. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.
14. Примеры рекурсивных функций: количество цифр в записи данного натурального числа; поиск минимального элемента в массиве; проверка, является ли число простым.
15. Внешние подпрограммы.
16. Этапы решения задач на компьютере.
17. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
18. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
19. Понятие величины. Типы величин. Присваивание величин. Совместимость по присваиванию.
20. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
21. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
22. Интегрированная среда разработки приложений.
23. Структура программы, элементы языка (алфавит). Понятие типа данных.
24. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
25. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
26. Условный оператор. Оператор множественного ветвления (выбора).
27. Циклы: с предусловием, с постусловием. Связь с другими циклами.
28. Циклы: с параметром. Связь с другими циклами.
29. Структурированные типы данных. Линейные массивы. Примеры задач.
30. Структурированные типы данных. Двумерные массивы. Примеры задач.
31. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
32. Сортировка массивов. Метод обмена.
33. Сортировка массивов. Метод вставок.
34. Подпрограммы. Основные способы передачи параметров в подпрограмму, их сравнение.
35. Подпрограммы. Область видимости. Локальные и глобальные идентификаторы.
36. Процедуры. Организация и вызов. Примеры.
37. Функции. Организация и вызов. Примеры.
38. Простые типы данных.

39. Структурированные типы данных. Строковый тип данных: основные процедуры и функции, примеры.
40. Рекурсия. Механизм рекурсии. Примеры.
41. Сортировка массивов. Метод быстрой сортировки.
42. Множества. Примеры.
43. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. Программирование типовых алгоритмов обработки записей.
44. Файловые типы. Общие процедуры для работы с файлами. Компонентные (типизированные) файлы.
45. Текстовые файлы. Текст-ориентированные процедуры и функции. Типовые задачи.
46. Прямой и последовательный доступ к компонентам файла. Процедуры и функции, ориентированные на прямой доступ к компонентам файла.
47. Поиск в типизированных файлах. Сортировка файлов (на примере одного из методов).
48. Типизированные файлы. Файлы записей. Типовые алгоритмы обработки.
49. Статическая и динамически распределяемая память. Пример использования указателей.
50. Динамические структуры данных. Однонаправленный список. Процедуры обработки списка.
51. Динамические структуры данных. Двухнаправленный список. Процедуры обработки списка.
52. Динамические структуры данных. Кольцевой список (однонаправленный или двухнаправленный). Процедуры обработки списка.
53. Модуль. Общая структура модуля. Компиляция и подключение модуля.
54. Объектно-ориентированное программирование. Компоненты.
55. Реализация индивидуального проекта ООП

Критерии оценивания

«5» «отлично» или «зачтено» – студент показывает глубокое и полное овладение содержанием программного материала по УД, в совершенстве владеет понятийным аппаратом и демонстрирует умение применять теорию на практике, решать различные практические и профессиональные задачи, высказывать и обосновывать свои суждения в форме грамотного, логического ответа (устного или письменного), а также высокий уровень овладения общими и профессиональными компетенциями и демонстрирует готовность к профессиональной деятельности;

«4» «хорошо» или «зачтено» – студент в полном объеме освоил программный материал по УД владеет понятийным аппаратом, хорошо

ориентируется в изучаемом материале, осознанно применяет знания для решения практических и профессиональных задач, грамотно излагает ответ, но содержание, форма ответа (устного или письменного) имеют отдельные неточности, демонстрирует средний уровень овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности;

«3» «удовлетворительно» или «зачтено» – студент обнаруживает знание и понимание основных положений программного материала по УД, но излагает его неполно, непоследовательно, допускает неточности в определении понятий, в применении знаний для решения практических и профессиональных задач, не умеет доказательно обосновать свои суждения, но при этом демонстрирует низкий уровень овладения общими и профессиональными компетенциями и готовность к профессиональной деятельности;

«2» «неудовлетворительно» или «не зачтено» – студент имеет разрозненные, бессистемные знания, не умеет выделять главное и второстепенное, допускает ошибки в определении понятий, беспорядочно и неуверенно излагает программный материал по УД, не умеет применять знания для решения практических и профессиональных задач, не демонстрирует овладение общими и профессиональными компетенциями и готовность к профессиональной деятельности.

3. Информационное обеспечение

перечень учебных изданий, электронных изданий, электронных и Интернет-ресурсов, образовательных платформ, электронно-библиотечных систем, веб-систем для организации дистанционного обучения и управления им, используемые в образовательном процессе как основные и дополнительные источники.

Основные источники:

1. Алгоритмизация и программирование, учебное пособие/ Канцедал С.А. – М.: ИД ФОРУМ,2017-352
2. Основы алгоритмизации и программирования (1-е изд.) учебник/ Семакин И.Г. – М.:ИЦ Академия,2017 – 304 с.
3. Основы программирования. Учебник и практикум для СПО Черпаков И.В. .- М.: Юрайт,2018-219 с.
4. Черпаков И.В. Основы программирования: Учебник и практикум для СПО.- М.: Юрайт,2018.-219 с.

Дополнительные источники:

5. Богомазова Г.Н. Модернизация программного обеспечения персональных компьютеров, серверов, периферийных устройств и оборудования: учебник. – М.: ИЦ «Академия», 2016. – 192 с.

- Галицына О.Л. Языки программирования: учебное пособие / О.Л. Галицына, Т.Л. Партыка, И.И. Попов. – 3-е изд., перераб. и доп. – М.: ФОРУМ; ИНФРА-М, 2013. – 400 с.: ил. – (Профессиональное образование).
6. Голицына, Т.Л. Партыка, И.И. Попов. – 3-е изд., перераб. и доп. – М.: ФОРУМ; ИНФРА-М, 2013. – 400 с.: ил. – (Профессиональное образование).
 7. Канцедал С.А. Алгоритмизация и программирование: учебное пособие – М.: ИД «ФОРУМ»: ИНФРА-М, 2010.
 8. Кормен Т. Алгоритмы: вводный курс.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2014.
 9. Моргун А.Н. Программирование на языке Pascal. Основы обработки структур данных. – М.: «Вильямс», 2006.
 - 10 Рыбальченко М.В. Архитектура информационных систем. Учебное пособие для СПО / М.В. Рыбальченко – М. ИЦ «Юрайт», 2017 – 91 с.
 - 11 Рапаков Г.Г. Turbo Pascal для студентов и школьников / Г.Г. Рапаков, С.Ю. Ржеуцкая. – СПб.: БХВ-Петербург, 2013. – 352 с.: ил.
 - 12 Семакин И.Г., Шестаков А.П. / Основы алгоритмизации и программирования. Практикум: учебн. пособие для студ. учреждений СПО – М.: Издательский центр «Академия», 2014.
- Электронные издания (электронные ресурсы):**
- 13 CNews [Электронный ресурс] / Официальный сайт периодического издания – журнал «CNews». Режим доступа: <http://www.cnews.ru>, свободный.
 - 14 Computerworld – Россия [Электронный ресурс] / Официальный сайт периодического издания – международный компьютерный журнал «Computerworld». Режим доступа: <http://www.computerworld.ru>, свободный.
 - 15 PascalABC.NET [Электронный ресурс] / Современное программирование на языке Паскаль. – Режим доступа: <http://pascalabc.net>, свободный.
 - 16 VisualBasic.NET [Электронный ресурс] / Мультимедийный Обучающий Курс. – Режим доступа: <http://gratisfile.com/publ/140-1-0-13>, свободный.
 - 17 Википедия – свободная энциклопедия [Электронный ресурс] / Сайт международного информационного ресурса «Википедия» – Режим доступа: <http://ru.wikipedia.org>, свободный.
 - 18 Компьютерные видео уроки. [Электронный ресурс] / Компьютерные видео уроки по программированию. – Режим доступа: <http://compteacher.ru/programming>, свободный.
 - 19 Мир ПК [Электронный ресурс] / Официальный сайт периодического издания – журнал «Мир ПК». – Режим доступа: <http://www.osp.ru/pcworld/#/home>, свободный.
 - 20 Открытые системы. СУБД [Электронный ресурс] / Официальный сайт периодического издания – журнал «Открытые системы. СУБД». – Режим доступа: <http://www.osp.ru/os/#/home>, свободный.
 - 21 Программные продукты и системы [Электронный ресурс] / Официальный сайт периодического издания – журнал «Программные продукты и системы». – Режим доступа: <http://www.swsys.ru>, свободный.

- 22 Федеральный центр информационно-образовательных ресурсов [Электронный ресурс] / Информационный сайт. – Режим доступа: <http://fcior.edu.ru>.
- 23 Язык Pascal [Электронный ресурс] / Программирование для начинающих. – Режим доступа: <http://www.pas1.ru>, свободный.
Цифровая образовательная среда СПО PROобразование:
Алексеев, Е. Р. Free Pascal и Lazarus : учебник по программированию / Е. Р. Алексеев, О. В. Чеснокова, Т. В. Кучер. — 2-е изд. — Саратов : Профобразование, 2019. — 438 с. — ISBN 978-5-4488-0105-1. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROобразование : [сайт]. — URL: <https://profspo.ru/books/87979> (дата обращения: 04.09.2020). — Режим доступа: для авторизир. пользователей
Цифровая образовательная среда СПО PROобразование:
Златопольский, Д. М. Программирование: типовые задачи, алгоритмы, методы / Д. М. Златопольский. — 4-е изд. — Москва : Лаборатория знаний, 2020. — 224 с. — ISBN 978-5-00101-789-9. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROобразование : [сайт]. — URL: <https://profspo.ru/books/12264> (дата обращения: 03.09.2020). — Режим доступа: для авторизир. пользователей
Цифровая образовательная среда СПО PROобразование:
Кудинов, Ю. И. Основы алгоритмизации и программирования : учебное пособие для СПО / Ю. И. Кудинов, А. Ю. Келина. — 2-е изд. — Липецк, Саратов : Липецкий государственный технический университет, Профобразование, 2020. — 71 с. — ISBN 978-5-88247-956-4, 978-5-4488-0757-2. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROобразование : [сайт]. — URL: <https://profspo.ru/books/92834> (дата обращения: 07.09.2020). — Режим доступа: для авторизир. пользователей
Цифровая образовательная среда СПО PROобразование:
Лубашева, Т. В. Основы алгоритмизации и программирования : учебное пособие / Т. В. Лубашева, Б. А. Железко. — Минск : Республиканский институт профессионального образования (РИПО), 2016. — 379 с. — ISBN 978-985-503-625-9. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROобразование : [сайт]. — URL: <https://profspo.ru/books/67689> (дата обращения: 06.09.2020). — Режим доступа: для авторизир. пользователей

- Цифровая образовательная среда СПО PROФобразование:
Мухаметзянов, Р. Р. Основы программирования в Delphi : учебно-методическое пособие / Р. Р. Мухаметзянов. — Набережные Челны : Набережночелнинский государственный педагогический университет, 28 2017. — 137 с. — ISBN 2227-8397. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/66811> (дата обращения: 05.09.2020). — Режим доступа: для авторизир. пользователей
- Цифровая образовательная среда СПО PROФобразование:
Токманцев, Т. Б. Алгоритмические языки и программирование : учебное пособие для СПО / Т. Б. Токманцев ; под редакцией В. Б. Костоусова. — 2-е изд. — Саратов, Екатеринбург : Профобразование, Уральский 29 федеральный университет, 2019. — 102 с. — ISBN 978-5-4488-0510-3, 978-5-7996-2899-4. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/87785> (дата обращения: 01.09.2020). — Режим доступа: для авторизир. Пользователей
- Цифровая образовательная среда СПО PROФобразование:
Уйманова, Н. А. Основы объектно-ориентированного программирования : практикум для СПО / Н. А. Уйманова, М. Г. 30 Таспаева. — Саратов : Профобразование, 2019. — 155 с. — ISBN 978-5-4488-0352-9. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/86199> (дата обращения: 04.09.2020). — Режим доступа: для авторизир. пользователей
- Цифровая образовательная среда СПО PROФобразование:
Устинов, В. В. Основы алгоритмизации и программирование. Часть 2 : конспект лекций / В. В. Устинов. — Новосибирск : Новосибирский 31 государственный технический университет, 2013. — 32 с. — ISBN 978-5-7782-2337-0. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/44675> (дата обращения: 06.09.2020). — Режим доступа: для авторизир. пользователей
- Цифровая образовательная среда СПО PROФобразование:
Устинов, В. В. Основы алгоритмизации и программирования. Часть 1 : конспект лекций / В. В. Устинов. — Новосибирск : Новосибирский 32 государственный технический университет, 2010. — 40 с. — ISBN 978-5-7782-1366-1. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/44676> (дата обращения: 07.09.2020). — Режим доступа: для авторизир. пользователей

- Цифровая образовательная среда СПО PROФобразование:
Чурина, Т. Г. Основы алгоритмизации и программирования : учебное пособие для СПО / Т. Г. Чурина, Т. В. Нестеренко. — Саратов, Москва : Профобразование, Ай Пи Ар Медиа, 2020. — 214 с. — ISBN 978-5-4488-0802-9, 978-5-4497-0465-8. — Текст : электронный // Электронный ресурс цифровой образовательной среды СПО PROФобразование : [сайт]. — URL: <https://profspo.ru/books/96017> (дата обращения: 03.09.2020). — Режим доступа: для авторизир. пользователей
- Электронно-библиотечная система:**
IPR BOOKS - <http://www.iprbookshop.ru/78574.html>
- Веб-система для организации дистанционного обучения и управления им:**
Система дистанционного обучения ОГАПОУ «Алексеевский колледж»
<http://moodle.alcollege.ru/>