

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Институт *высокоточных систем им. В.П. Грязева*

Кафедра «*Ракетное вооружение*»

Утверждено на заседании кафедры
«*Ракетное вооружение*»
« 15 » 01 2020 г., протокол № 5

И.о. зав. кафедрой

 В.А. Никитин

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫЕ МАТЕРИАЛЫ) ДЛЯ
ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ (МОДУЛЮ)
«*Информатика*»»**

**основной профессиональной образовательной программы
высшего образования – программы бакалавриата**

по направлению подготовки
24.03.03 Баллистика и гидроаэродинамика

с направленностью (профилем)
Баллистика ракет и снарядов

Форма обучения: *очная*

Идентификационный номер образовательной программы: 240303-01-20

Тула 2020 год

ЛИСТ СОГЛАСОВАНИЯ
фонда оценочных средств (оценочных материалов)

Разработчик:

Конюхова Л.А., доцент, к.т.н.

(ФИО, должность, ученая степень, ученое звание)



(подпись)

1. Описание фонда оценочных средств (оценочных материалов)

Фонд оценочных средств (оценочные материалы) включает в себя контрольные задания и (или) вопросы, которые могут быть предложены обучающемуся в рамках текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю). Указанные контрольные задания и (или) вопросы позволяют оценить достижение обучающимся планируемых результатов обучения по дисциплине (модулю), установленных в соответствующей рабочей программе дисциплины (модуля), а также сформированность компетенций, установленных в соответствующей общей характеристике основной профессиональной образовательной программы.

Полные наименования компетенций и индикаторов их представлены в общей характеристике основной профессиональной образовательной программы.

2. Оценочные средства (оценочные материалы) для проведения текущего контроля успеваемости обучающихся по дисциплине (модулю)

1 семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. Укажите алгоритмический язык высокого уровня, предназначенный для написания программ, выполняемых виртуальной машиной на компьютерах любой архитектуры, и содержащий дополнительные элементы, ориентированные на программирование для сети Internet: команды доступа к узлу Internet и специфические элементы работы с файловыми системами в масштабах Internet.

1. C++.
2. Pascal.
3. Java.
4. Fortran.
5. Basic

2. Для чего предназначен язык HTML?

1. Язык гипертекстовой разметки для создания простых приложений (статический текст на экране или текст с гиперсвязями, текст с графическими картинками).
2. Язык для создания сложных приложений, связанных с математическими расчетами, работающих в сети Internet.
3. Язык для создания графических и текстовых редакторов, работающих в сети Internet.
4. Язык для управления работой удаленных компьютеров в сети Internet.
5. Язык для создания любых приложений, работающих в сети Internet.

3. Что такое "полиморфизм":

1. способность создания нескольких функций с одинаковым именем, различающихся аргументами;
2. способность создания нескольких классов с одинаковым именем, различающихся функциями;
3. способность создания нескольких массивов с одинаковым именем, различающихся размерностью;
4. способность создания нескольких структур с одинаковым именем, различающихся функциями;

5. способность создания нескольких переменных с одинаковым именем, различающихся типом;

4. С помощью какого выражения можно открыть файл:

1. `#include<iostream.h>`
2. `#include<stdio.h>`
3. `fscanf(dan, "%f", &x);`
4. `fopen(dan);`
5. `fclose(dan)`

5. С помощью оператора `new`:

1. Осуществляют выделение памяти под динамический массив.
2. Освобождают память, выделенную под динамический массив.
3. Начинают новую страницу программы.
4. Описывают начало цикла.
5. Обновляют содержимое массива.

6. Что такое "инкапсуляция"?

1. объединение переменных в функции;
2. объединение переменных и аргументов в функции;
3. объединение данных и функций в классе;
4. объединение функций в программе;
5. объединение данных и функций в программе.

7. Что такое "наследование":

1. передача параметров в функцию через ее аргументы;
2. возвращение параметров из функции через ее аргументы;
3. способность создания функции на базе другой функции;
4. способность создания класса на базе другого класса;
5. способность создания программы на базе другой программы.

8. Укажите правильное объявление конструктора класса `class CVector`:

1. `Constructor :: CVector();`
2. `CVector :: CVector();`
3. `Constructor : CVector();`
4. `CVector : CVector();`
5. `CVector->CVector();`

9. Укажите правильное объявление деструктора класса `class CVector`:

1. `Destructor :: CVector();`
2. `~CVector :: CVector();`
3. `Denstructor : CVector();`
4. `CVector :: ~CVector();`
5. `CVector : ~CVector();`

10. В каком месте программы могут располагаться функции?

1. До функции `main ()`
2. После функции `main ()`
3. В любом месте
4. Внутри функции `main ()`
5. Вне функции `main ()`

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. В качестве критерия выбора наилучшего алгоритма используют:

1. *Оценку точности решения задачи*
2. *Затраты времени на ее решение*
3. *Некоторый интегральный критерий, включающий оценки и точности, и затраты времени*
4. *Минимальное количество операторов*
5. *Любой из 1, 2, 3*

2. Библиотека `<iostream.h>` используется для:

1. *Для организации ввода-вывода в программе*
2. *Для возможности использовать математические функции*
3. *Для организации только ввода в программе*
4. *Для организации только вывода в программе*
5. *Для возможности применения функции `Close ()`*

3. Как осуществляется автоматический выбор шага, обеспечивающий получение результата с заданной погрешностью решения?

1. *Коррекцией расчета по конечному полученному значению функции.*
2. *Коррекцией расчета по текущей величине получаемой на каждом шаге первой производной функции.*
3. *Коррекцией расчета по текущей величине интеграла получаемой функции.*
4. *Коррекцией расчета по текущей величине получаемой на каждом шаге второй производной функции.*
5. *Коррекцией расчета по текущему получаемому значению функции на каждом шаге интегрирования.*

4. Что указывает ключевое слово "protected"?

1. *программа может быть использована только одним пользователем;*
2. *операторы, объединенные этим ключевым словом, могут быть использованы только функциями данного модуля программы;*
3. *аргументы функции не общедоступны;*
4. *операторы, объединенные этим ключевым словом, могут быть использованы только функциями данного класса;*
5. *операторы, объединенные этим ключевым словом, могут быть использованы только функциями данного класса и классов-потомков;*

5. Под функцией или процедурой понимается

1. *Законченная последовательность арифметических действий многократно выполняющихся в программе.*
2. *Законченная последовательность логических действий многократно выполняющихся в программе.*
3. *Законченная последовательность арифметических действий и незаконченная последовательность логических действий многократно выполняющихся в программе.*
4. *Незаконченная последовательность арифметических действий и законченная последовательность логических действий многократно выполняющихся в программе.*
5. *Законченная последовательность арифметических и логических действий, выполняющихся в программе.*

6. Что произойдет если пропустить инструкцию `break` при использовании оператора `switch`

1. Компьютер не станет выполнять оператор `switch`
2. Компьютер выполнит все инструкции, помещенные в соответствующую ветви `case`, и прекратит выполнение оператора `switch`
3. Компьютер выполнит все инструкции, помещенные в соответствующей ветви `case`, и далее, вплоть до первого встреченного в тексте `break`
4. Компьютер выполнит все инструкции, помещенные в соответствующей ветви `case`, и прекратит выполнение оператора `switch`
5. Компьютер перейдет к выполнению инструкции `default`

7. Унарная логическая операция `&` применима:

1. К переменным
2. К элементам массива
3. И к переменным, и к элементам массива
4. Как логическое «и»
5. К функции

8. Двумерный массив описывается следующим образом (в скобках указаны константы)

1. `double A (i) (i)`
2. `int A [i] [i]`
3. `double A {} {}`
4. `double A[i x i]`
5. `int A[i,i]`

9. С помощью оператора `new`:

1. Осуществляют выделение памяти под динамический массив.
2. Освобождают память, выделенную под динамический массив.
3. Начинают новую страницу программы.
4. Описывают начало цикла.
5. Обновляют содержимое массива.

10. После окончания работы с динамическим массивом необходимо освободить выделенную память с помощью оператора:

1. `delete[] x;`
2. `Close ();`
3. `break;`
4. `default;`
5. `case()`.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. Редактор форм:

1. Предназначен для создания интерфейса программы с пользователем.
2. Предназначен для написания текста программы, в частности, функций обработки событий.
3. Позволяет путем программирования устанавливать свойства объектов без необходимости рутинного программирования.
4. Содержит события, которые можно связывать с кодами реакции объектов на их возникновение.
5. Содержит объекты, которые разделяются многими приложениями с целью уменьшения временных затрат при разработке.

2. На каких элементах построены ЦВМ второго поколения:

1. На интегральных схемах.
2. На интегральных подсистемах.
3. На дискретных полупроводниках и магнитных элементах.
4. На электровакуумных элементах.
5. На электромеханических элементах.

3. На решение каких задач ориентирован алгоритмический язык ФОРТРАН?

1. Для написания программ выполнения экономических расчетов.
2. Для написания программ, являющихся операционными системами.
3. Для написания программ, являющихся играми.
4. Для написания программ, являющихся драйверами устройств.
5. Для написания программ выполнения математических расчетов.

4. Инспектор объектов

1. Предназначен для создания интерфейса программы с пользователем.
2. Предназначен для написания текста программы, в частности, функций обработки событий.
3. Позволяет визуально устанавливать свойства объектов без необходимости рутинного программирования и содержит события, которые можно связывать с кодами реакции объектов на их возникновение.
4. Содержит объекты, которые разделяются многими приложениями с целью уменьшения временных затрат при разработке.
5. Предназначен для организации диалога программы с пользователем.

5. Как осуществляется автоматический выбор шага, обеспечивающий получение результата с заданной погрешностью решения?

1. Коррекцией расчета по конечному полученному значению функции.
2. Коррекцией расчета по текущей величине получаемой на каждом шаге первой производной функции.
3. Коррекцией расчета по текущей величине интеграла получаемой функции.

4. Коррекцией расчета по текущей величине получаемой на каждом шаге второй производной функции.
5. Коррекцией расчета по текущему получаемому значению функции на каждом шаге интегрирования.

6. На каких элементах построены ЦВМ четвертого поколения:

1. На интегральных схемах.
2. На интегральных подсистемах.
3. На дискретных полупроводниках и магнитных элементах.
4. На электровакуумных элементах.
5. На электромеханических элементах.

7. Оператор `while ()` является:

1. Оператором условия.
2. Оператором выбора.
3. Оператором цикла.
4. Оператором прерывания.
5. Оператором выхода.

8. Оператор `for ()` является:

1. Оператором условия.
2. Оператором выбора.
3. Оператором цикла.
4. Оператором прерывания.
5. Оператором выхода.

9. Компонент вкладки `Standard TLabel`

1. Создает панель команд главного меню для формы.
2. Отображает на форме текст названия.
3. Создает кнопку с надписью.
4. Отображает область списка текстовых строк.
5. Отображает область редактируемого ввода одиночной строки информации на форме.

10. Какой компонент вкладки `Additional` создает графическую кнопку быстрого вызова?

1. `TSpeedButton`
2. `TShape`
3. `TStringGrid`
4. Никакой из перечисленных.
5. Любой из перечисленных.

2 семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. Символ `&`:

1. Обозначает логическое «и»
2. Указывает на адрес переменной или элемент массива
3. Является признаком начала директивы препроцессора
4. Указывает на выделение объема памяти под динамический массив

5. Символ, часто используемый в именах переменных

2. Какой оператор цикла дает возможность включить инициализирующее выражение, исполняемое один раз перед тем, как будет произведена оценка условия, и указать выражение, которое будет исполняться после каждой итерации оператора или блока, связанного с циклом?

1. *do/while*
2. *for*
3. *while*
4. *Все*
5. *for и while*

3. Что будет получено при поиске экстремума в результате выполнения итерационного процесса метода деления интервала поиска, если в области поиска нет локального экстремума?

1. Произойдет "защелкивание".
2. Процесс сойдется к середине интервала (a, b) .
3. Процесс сойдется к точке a .
4. Процесс сойдется к точке b .
5. Процесс сойдется к одной из границ интервала.

4. Как осуществляется автоматический выбор шага, обеспечивающий получение результата с заданной погрешностью решения?

1. Коррекцией расчета по конечному полученному значению функции.
2. Коррекцией расчета по текущей величине получаемой на каждом шаге первой производной функции.
3. Коррекцией расчета по текущей величине интеграла получаемой функции.
4. Коррекцией расчета по текущей величине получаемой на каждом шаге второй производной функции.
5. Коррекцией расчета по текущему получаемому значению функции на каждом шаге интегрирования.

5. Что такое алгоритм решения задачи?

1. Последовательность арифметических действий.
2. Последовательность логических действий.
3. Последовательность операторов.
4. Последовательность арифметических и логических действий.
5. Последовательность циклических действий.

6. Укажите действие, которое не выполняется с помощью модема.

1. Посылка электронного письма.
2. Посылка факса.
3. Проведение телеконференции в сети Интернет.
4. Работа с удаленной базой данных.
5. Вывод изображения на экран монитора.

7. В чем состоит назначение сети "Интернет"?

1. Сеть для объединения удаленных компьютеров по схеме "Звезда" и архитектуре "Клиент-Сервер".
2. Сеть для объединения удаленных компьютеров по схеме "Паутина".
3. Сеть для объединения локальных компьютеров по одно-ранговой схеме.

4. Сеть для объединения локальных компьютеров по схеме "Звезда" и архитектуре "Клиент-Сервер".
5. Объединение локальных компьютеров с помощью LPT- или COM-порта.

8. Редактор форм:

1. Предназначен для создания интерфейса программы с пользователем.
2. Предназначен для написания текста программы, в частности, функций обработки событий.
3. Позволяет путем программирования устанавливать свойства объектов без необходимости рутинного программирования.
4. Содержит события, которые можно связывать с кодами реакции объектов на их возникновение.
5. Содержит объекты, которые разделяются многими приложениями с целью уменьшения временных затрат при разработке.

9. Что такое алгоритм линейной структуры (линейный алгоритм)?

1. Алгоритм, в котором предусмотрено неоднократное исполнение одной и той же последовательности действий.
2. Алгоритм, в котором предусмотрено разветвление выполняемой последовательности действий в зависимости от результата проверки какого-либо условия.
3. Алгоритм, в котором последовательно расположены циклические и логические действия.
4. Алгоритм, в котором последовательно расположены арифметические действия.
5. Алгоритм, в котором блоки выполняются последовательно друг за другом.

10. В качестве критерия выбора наилучшего алгоритма используют:

1. Оценку точности решения задачи
2. Затраты времени на ее решение
3. Некоторый интегральный критерий, включающий оценки и точности, и затраты времени
4. Минимальное количество операторов
5. Любой из 1, 2, 3

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1 Укажите неправильное имя класса:

1. CStud
2. CStudent
3. CStud21
4. CStud_21
5. CStud-21

2. Укажите стандартную библиотеку, необходимую для использования функций $\sin(x)$, $\exp(x)$, \sqrt{x} , $\log(x)$, $\text{abs}(x)$:

1. <stdio.h>
2. <iostream.h>
3. <math.h>
4. <conio.h>
5. <string.h>

3. Укажите оператор, не содержащий ошибки, вычисляющий корень квадратного уравнения ax^2+bx+c оптимально с точки зрения быстродействия:

1. $x1 = (-b + \sqrt{b*b - 4.0*a*c}) / 2.0*a;$
2. $x1 = (-b + \sqrt{\text{pow}(b,2) - 4.0*a*c}) / (2.0*a);$
3. $x1 = (-b + \text{pow}(\text{pow}(b,2) - 4.0*a*c), 0.5) / (2.0*a);$
4. $x1 = (-b + \sqrt{b*b - 4.0*a*c}) / (2.0*a);$
5. $x1 = -b + \sqrt{b*b - 4.0*a*c} / (2.0*a);$

4. Укажите логический оператор:

1. `switch (c)`
2. `goto m1`
3. `for (i=0; i>10; i++)`
4. `if (i > 0)`
5. `while (i > 0)`

5. Что такое "массив"?

1. множество описаний переменных, объединенных скобками `{}`;
2. операторы, объединенные скобками `{}`;
3. множество переменных разных типов, объединенных ключевым словом `class`;
4. множество переменных одного типа, объединенных одним именем;
5. множество строк программы.

6. С помощью оператора `new`:

6. Осуществляют выделение памяти под динамический массив.
7. Освобождают память, выделенную под динамический массив.
8. Начинают новую страницу программы.
9. Описывают начало цикла.
10. Обновляют содержимое массива.

7. После окончания работы с динамическим массивом необходимо освободить выделенную память с помощью оператора:

6. `delete[] x;`
7. `Close ();`
8. `break;`
9. `default;`
10. `case().`

8. По какому признаку различают алгоритмы линейный, разветвляющийся и циклический:

1. По количеству переменных
2. По количеству блоков
3. По характеру связей между блоками
4. В зависимости от вида функции
5. Такого разделения не существует

9. Укажите правильное объявление ссылки.

1. `int x; int a=*x;`
2. `int x; int &a=&x;`
3. `int x; int a=&x;`
4. `int x; int a=x;`
5. `int x; int &a=x;`

10. Что означает символ `;` после метки `m1`; ?

1. Конец оператора метка.

2. Пустой оператор.
3. Конец группы операторов.
4. Необходимость продолжения цикла.
5. Конец оператора *switch*.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. Укажите правильный досрочный выход из всех циклов для продолжения выполнения программы.

1. `for (int i=0; i<20; i++)
for (int j=0; j<10; j++) { if(y[i][j] > 2.0)continue; y[i][j] = 1.0; }`
2. `for (int i=0; i<20; i++)
for (int j=0; j<10; j++) { if(y[i][j] > 2.0)exit; y[i][j] = 1.0; }`
3. `for (int i=0; i<20; i++)
for (int j=0; j<10; j++) { if(y[i][j] > 2.0)abort; y[i][j] = 1.0; }`
4. `for (int i=0; i<20; i++)
for (int j=0; j<10; j++) { if(y[i][j] > 2.0)goto m1; y[i][j] = 1.0; }`
5. `for (int i=0; i<20; i++)
for (int j=0; j<10; j++) { if(y[i][j] > 2.0)break; y[i][j] = 1.0; }`

2. Укажите правильное объявление ссылки.

1. `int x; int a=*x;`
2. `int x; int &a=&x;`
3. `int x; int a=&x;`
4. `int x; int a=x;`
5. `int x; int &a=x;`

3. Какое логическое выражение соответствует неравенствам : $x < a$ или $x > b$?

1. `a > x && x < b`
2. `a > x || x < b`
3. `a < x && x < b`
4. `a > x || x > b`
5. `a < x && x > b`

4. Укажите правильный результат операции z ?

```
double x=1.5, y=3.0, z=1.0;
if ( x<1 || x>2 && y<1 || y>2)z=0.5;
else z=1.5;
```

1. 1.5;
2. 3.0;
3. 1.0;
4. 0.5;
5. 2;

5. В чем отличие описаний ключевыми словами "class" и "struct"?

1. в способах наследования;
2. в типе доступа к данным и функциям;
3. в способах передачи аргументов функциям при наследовании;
4. в способе наследования;
5. в типе полиморфизма.

6. Укажите правильный оператор, если есть описание $\text{int } i, j; \text{ double } x[10], y;$

1. $x[y] = 1.5;$
2. $x[3/2] = y;$
3. $x[j+y] = j;$
4. $x[j] = j+y;$
5. $x[y/j] = 1.5;$

7. Назначение ключевого слова "class"?

1. для описания полиморфизма;
2. для объединения операторов заданной функции;
3. для объединения данных и функций в единый объект;
4. для классификации переменных по типам;
5. для указания наследования функции.

8. Укажите правильное объявление деструктора класса $\text{class } CVector:$

6. $\text{Destructor} :: CVector();$
7. $\sim CVector :: CVector();$
8. $\text{Denstructor} : CVector();$
9. $CVector :: \sim CVector();$
10. $CVector : \sim CVector();$

9. С помощью какого выражения можно закрыть файл:

1. `#include<iostream.h>`
2. `#include<stdio.h>`
3. `fscanf(dan,"%f",&x);`
4. `fopen(dan);`
5. `fclose(dan);`

10. Что означает выражение $i++$; ?

1. Сдвиг переменной i на один бит после использования ее в ходе выполнения программы.
2. Увеличение переменной i на единицу перед использованием ее в ходе выполнения программы.
3. Увеличение переменной i на единицу после использования ее в ходе выполнения программы.
4. Умножение переменной i на два после использования ее в ходе выполнения программы.
5. Умножение переменной i на два перед использованием ее в ходе выполнения программы.

1. Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

1 семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. *Этапы развития вычислительной техники.*
2. *Виды погрешностей решения задач на ЭВМ.*
3. *История развития языков программирования.*
4. *Классификация языков программирования.*

5. *Основные принципы объектно-ориентированного программирования*
6. *Понятия информатики и информации*
7. *Особенности программирования на языках высокого уровня*
8. *Основные операторы языка C++.*
9. *Основные составные части ПК.*
10. *Устройства ввода-вывода для ПК.*

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. *Основы программирования на языках высокого уровня*
2. *Основы работы в среде визуального программирования*
3. *Ввод и вывод информации*
4. *Основные операторы языка программирования высокого уровня*
5. *Основы работы с массивами данных.*
6. *Логические операторы*
7. *Сравнительный анализ операторов цикла.*
8. *Особенности использования оператора выбора.*
9. *Правила задания имен переменных.*
10. *Особенности построения графиков в C++*

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. *Основные принципы создания эффективных программ.*
2. *Основные принципы работы с файлами.*
3. *Основы использования обработчиков элементов ввода-вывода информации в среде программирования*
4. *Основы использования элементов вывода графической информации в среде программирования*
5. *Программные средства реализации информационных процессов.*
6. *Основы работы с ключевыми словами "private" и "protected".*
7. *Опишите стандартные библиотеки, необходимые для работы со стандартными строками типа *char.*
8. *Что такое процедура или функция в программировании?*
9. *Для чего служат редактор кода и инспектор объектов?*
10. *Погрешности машинных вычислений.*

2семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. *Виды моделирования.*
2. *Модели объектов и процессов.*
3. *Локальные и глобальные сети.*
4. *Базы данных.*
5. *Файловые системы.*
6. *Основы алгоритмизации.*
7. *Виды алгоритмов.*
8. *Основные свойства информации.*
9. *Сетевые технологии хранения информации.*
10. *Архитектура баз данных.*

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. *Основы информационной безопасности.*
2. *Основы работы с массивами*
3. *Методы и средства защиты информации*
4. *Основы построения алгоритмов.*
5. *Виды сетей и их описание.*
6. *Разработка и проектирование баз данных*
7. *Глобальная сеть Интернет.*
8. *Основные этапы решения инженерных задач*
9. *Структурированные типы.*
10. *Конструктор и деструктор.*

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. *Использование оператора условия при решении прикладных задач*
2. *Использование оператора выбора при решении прикладных задач*
3. *Использование операторов цикла при решении прикладных задач*
4. *Использование массивов при решении прикладных задач*
5. *Вывод графической информации*
6. *Характерные приемы программирования при решении инженерных задач*
7. *Расчет траектории полета*
8. *Построение алгоритма расчета траектории полета ЛА*
9. *Практическое использование средств защиты информации.*
10. *Свойства локальных и глобальных сетей.*

2. Оценочные средства (оценочные материалы) для проведения промежуточной аттестации обучающихся (защиты курсовой работы) по дисциплине (модулю)

2 семестр

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.1)

1. *Математическая модель расчета траектории полета*
2. *Численная модель расчета траектории полета*
3. *Физическая модель расчета траектории полета*
4. *Физические допущения, используемые для расчета траектории полета*
5. *Этапы решения задач на компьютере*
6. *Построения алгоритма расчета траектории полета*
7. *Ввод и вывод данных (на примере расчета траектории полета).*
8. *Использование операторов цикла для расчета траектории полета*
9. *Использование операторов условия для расчета траектории полета*
10. *Использование стандартных библиотек, необходимых для расчета траектории полета*

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.2)

1. Численное решение обыкновенного дифференциального уравнения первого порядка методом Эйлера.
2. Уточнение численного решения обыкновенного дифференциального уравнения первого порядка методом Эйлера.
3. Алгоритм вычислительного процесса метода Рунге-Кутты четвертого порядка.
4. Вывод конечно-разностного выражения для второй производной.
5. Вывод конечно-разностных соотношений для решения задачи о траектории полета тела в воздухе.
6. Обоснование выбора численной модели для решения задачи расчета траектории полета
7. Вывод информации в файловой форме (на примере расчета траектории полета).
8. Вывод информации в графической форме (на примере расчета траектории полета).
9. Особенности объявления переменных при решении инженерных задач.
10. Решение системы дифференциальных уравнений методом Эйлера.

Перечень контрольных заданий и (или) вопросов для оценки сформированности компетенции ОПК-2 (контролируемый индикатор достижения компетенции ОПК-2.3)

1. Алгоритм расчета траектории полета тела в воздухе с использованием вторых производных.
2. Алгоритм расчета траектории полета тела в воздухе с использованием первых производных.
3. Конечно-разностная модель для расчета траектории полета тела в воздухе с использованием первых производных.
4. Конечно-разностная модель для расчета траектории полета тела в воздухе с использованием вторых производных.
5. Обеспечение точности решения обыкновенного дифференциального уравнения заданным численным методом?
6. Расчет траектории движения неуправляемого ЛА методом конечных разностей
7. Конечно-разностные представления производных
8. Повышение точности аппроксимации производных
9. Расчет координат траектории X и Y с использованием конечно-разностной схемы.
10. Вычисление проекций скорости V_x и V_y с использованием конечно-разностной схемы.