

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ЛУГАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ВЛАДИМИРА ДАЛЯ»

Факультет компьютерных систем и информационных технологий

Кафедра информационных и управляющих систем

УТВЕРЖДАЮ

Декан факультета компьютерных
систем и информационных
технологий



Кочевский А.А.

« 19 »

2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по учебной дисциплине

«Программное обеспечение мехатронных и робототехнических систем»

15.03.06 Мехатроника и робототехника

«Компьютерные технологии управления в робототехнике и мехатронике»

Разработчик:

доцент

Юрков Д.А.

ФОС рассмотрен и одобрен на заседании кафедры информационных и управляющих систем от «18» апреля 2023 г., протокол № 15

Заведующий кафедрой

информационных и управляющих систем

Горбунов А.И.

Луганск 2023 г.

**Паспорт
фонда оценочных средств по учебной дисциплине
«Программное обеспечение
мехатронных и робототехнических систем»**

**Перечень компетенций, формируемых в результате освоения
учебной дисциплины**

№ п/п	Код контролируемой компетенции	Формулировка контролируемой компетенции	Контролируемые темы учебной дисциплины, практики	Этапы формирования (семестр изучения)
1	ОПК-14	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	Тема 1. Введение в язык программирования C/C++ Тема 2. Общая структура программы на языке C/C++ Тема 3. Операторы ввода-вывода Тема 4. Основные понятия языка C/C++ Тема 5. Составление выражений Тема 6. Условный оператор. Оператор выбора Тема 7. Циклы Тема 8. Одномерные массивы Тема 9. Строки Тема 10. Многомерные массивы Тема 11. Указатели Тема 12. Пользовательские типы данных Тема 13. Пользовательские функции. Классы памяти Тема 14. Передача параметров в функции Тема 15. Передача функций как параметров Тема 16. Разбиение программы на модули Тема 17. Рекурсивные функции Тема 18. Шаблоны функций Тема 19. Пространство имён Тема 20. Системы визуального проектирования Тема 21. Современные концепции разработки программ	Начальный(1) Основной(2) Основной(3)

			Тема 22. Основные классы и панель элементов Тема 23. Особенности языка C# Тема 24. Класс для представления табличных данных Тема 25. Класс Form и диалоговые окна Тема 26. Графический инструментарий Тема 27. Объектно-ориентированное программирование Тема 28. Свойства класса Тема 29. Наследование Тема 30. Полиморфизм Тема 31. Разработка отказоустойчивых приложений	
2	ПК-3	Способен разрабатывать программное обеспечение изделий робототехники и мехатроники, в том числе детской и образовательной робототехники	Тема 1. Введение в язык программирования C/C++ Тема 2. Общая структура программы на языке C/C++ Тема 3. Операторы ввода-вывода Тема 4. Основные понятия языка C/C++ Тема 5. Составление выражений Тема 6. Условный оператор. Оператор выбора Тема 7. Циклы Тема 8. Одномерные массивы Тема 9. Строки Тема 10. Многомерные массивы Тема 11. Указатели Тема 12. Пользовательские типы данных Тема 13. Пользовательские функции. Классы памяти Тема 14. Передача параметров в функции Тема 15. Передача функций как параметров Тема 16. Разбиение программы на модули Тема 17. Рекурсивные функции Тема 18. Шаблоны функций	Начальный(1) Основной(2) Основной(3)

			Тема 19. Пространство имён Тема 20. Системы визуального проектирования Тема 21. Современные концепции разработки программ Тема 22. Основные классы и панель элементов Тема 23. Особенности языка С# Тема 24. Класс для представления табличных данных Тема 25. Класс Form и диалоговые окна Тема 26. Графический инструментарий Тема 27. Объектно-ориентированное программирование Тема 28. Свойства класса Тема 29. Наследование Тема 30. Полиморфизм Тема 31. Разработка отказоустойчивых приложений	
--	--	--	--	--

Показатели и критерии оценивания компетенций, описание шкал оценивания

№ п/п	Код контролируемой компетенции	Показатель оценивания (знания, умения, навыки)	Контролируемые темы учебной дисциплины	Наименование оценочного средства
1	ОПК-14	Знает базовые алгоритмы, подходы к написанию ПО; Умеет использовать специализированные библиотеки; Владеет навыками использования высокоуровневых ЯП.	Тема 1-31	Фронтальные и индивидуальные опросы; контрольные работы; Промежуточная аттестация (экзамен)
2	ПК-3	Способен к написанию программного кода для изделий мехатроники и робототехники с использованием языков программирования, определения и манипулирования данными;	Тема 1-31	Фронтальные и индивидуальные опросы; контрольные работы; Промежуточная аттестация (экзамен)

		Проверяет работоспособность и рефакторинг кода программного обеспечения для изделий мехатроники и робототехники.		
--	--	--	--	--

Фонды оценочных средств по дисциплине

«Программное обеспечение мехатронных и робототехнических систем»

Вопросы для фронтальных и индивидуальных опросов:

Семестр 1

1. Какие основные этапы включает в себя решение задач на компьютере?
2. Какие этапы компьютерного решения задач осуществляются без участия компьютера?
3. Что называют математической моделью объекта или явления?
4. Почему невозможно точное исследование поведения объектов или явлений?
5. Какие способы моделирования осуществляются с помощью компьютера?
6. Из каких последовательных действий состоит процесс разработки программы?
7. Что называется алгоритмом?
8. Какими основными свойствами должен обладать алгоритм?
9. Какие существуют способы описания алгоритмов?
10. Какими графическими символами принято изображать в схемах алгоритма?
11. В чем отличие циклической структуры с предусловием от циклической структуры с постусловием?
12. Что такое параметр цикла?
13. В чем отличие регулярной циклической структуры от итеративной?
14. Доказывает ли получение правдоподобного результата правильность программы?
15. Какие ошибки могут остаться не выявленными, если не провести проверку (просмотр, прокрутку) программы?
16. Чем тестирование программы отличается от её отладки?
17. Можно ли с помощью тестирования доказать правильность программы?
18. На какой стадии работы над программой вычисляются эталонные результаты тестов?
19. Назовите основные этапы процесса тестирования.

20. В чём заключается отличие синтаксических ошибок от семантических?
21. О чём свидетельствует отсутствие сообщений машины о синтаксических ошибках?
22. Какие разновидности ошибок транслятор не в состоянии обнаружить?
23. Какова структура программы на языке Си?
24. Зачем нужна директива `#include`?
25. Что такое `main()`?
26. Перечислите скалярные типы данных языка Си.
27. Что определяет тип данного?
28. Что такое `void`?
29. Что такое явное и неявное приведение типов? Как и когда оно используется?
30. Что такое константа? Найдите константы в набранных вами программах.
31. Что такое переменная?
32. Как проинициализировать переменную?
33. Чем отличается оператор от операции?
34. Чем отличаются унарные операции от бинарных?
35. Какие операции относятся к арифметическим? Каков приоритет каждой из них?
36. Каков порядок выполнения операций в случае их одинакового приоритета?
37. Как выполняется операция деления в случае целочисленных операндов и в случае, когда хотя бы один из операндов вещественный?
38. Что такое выражение?
39. Какое значение вычисляет операция присваивания?
40. В каком порядке выполняются присваивания в случае, если в выражении их несколько?
41. Как и зачем используются дополнительные операции присваивания?
42. Чем отличается префиксная форма операции инкремента или декремента от постфиксной?
43. Какие функции используются для ввода информации? Назовите их отличительные особенности.
44. Какие функции используются для вывода информации? Назовите их отличительные особенности.
45. Почему функции `scanf()` и `printf()` называются функциями форматного ввода и вывода? Как они работают?
46. Чем отличается управляющая строка функции `scanf()` от управляющей строки функции `printf()`?
47. Что такое спецификатор формата? Зачем он нужен?
48. Какие параметры указываются функции `scanf()` после управляющей строки? Сколько их должно быть?
49. Каковы последствия несоответствия типа считываемой функцией `scanf()` переменной спецификатору типа?

50. Какие параметры указываются функции `printf()` после управляющей строки? Сколько их должно быть?
51. Каковы последствия несоответствия типа выводимого функцией `printf()` значения спецификатору типа?
52. Что такое управляющие символы? Зачем они нужны? Приведите примеры.
53. Чем отличается условная операция от условного оператора?
54. Что такое полная и неполная форма условного оператора?
55. Может ли существовать неполная форма условной операции?
56. Нужно ли писать `"else"`, если при выполнении условия выполняется оператор `return`?
57. Выражения какого типа могут определять условия?
58. Какие значения выражения, определяющего условие, считаются истинными, а какие ложными?
59. Какие операции относятся к операциям отношения?
60. Чем отличается операция `"= ="` от операции `"=="`?
61. Какие операции относятся к логическим? Каков их приоритет?
62. Какой операцией можно заменить операцию `"&&"` ?
63. Какой операцией можно заменить операцию `"||"` ?
64. Чему может быть равно значение выражения отношения или логического выражения?
65. Как правильно сравнить на равенство вещественные числа?
66. Как правильно проверить вхождение значения в некоторый диапазон?
67. Как проверить некоторое целочисленное значение на равенство нулю?
68. Как проверить отличие целочисленного значения от нуля?
69. Когда применяется вложение условных операторов?
70. Как правильно записать вложенные условные операторы?
71. Что представляет собой оператор `switch`? Как им пользоваться?
72. Как записать оператор `switch` с помощью условных операторов?
73. Что такое цикл?
74. Какие виды циклов вы знаете? Какие операторы цикла существуют в языке Си?
75. Чем отличается цикл с предусловием от цикла с постусловием?
76. Когда необходимо использовать цикл с предусловием, а когда с постусловием? Приведите примеры.
77. Какие циклы с предусловием существуют в языке Си?
78. Сколько операторов содержит в себе тело цикла в языке Си?
79. Как правильно записать цикл с постусловием на языке Си?
80. Как задать бесконечный цикл? Зачем он нужен? Как из него выйти?
81. Каким должно быть значение выражения, определяющего условие выполнения цикла, для завершения цикла?
82. Каким должно быть значение выражения, определяющего условие выполнения цикла, для выполнения тела цикла?
83. К чему приведет неправильное задание выражения, определяющего

условие выполнения цикла?

84. Может ли тело цикла отсутствовать? Если может, то приведите примеры таких циклов.

85. Чем отличается оператор *while* от оператора *if*?

86. Каков порядок действий при выполнении цикла *for*?

87. Как организовать арифметический цикл с помощью цикла *for*?

88. Запишите алгоритм, определяемый циклом *for*, с помощью цикла *while*.

89. Что такое вложенный цикл?

90. Сколько раз в общей сложности выполняется тело вложенного цикла?

91. Как и когда используются операторы *break* и *continue*?

92. Что такое рекуррентные вычисления? Когда они используются? Как их программировать?

93. Что такое указатель?

94. Какой объем памяти занимает указатель?

95. Что является значением переменной-указателя?

96. Как проинициализировать указатель?

97. Что такое NULL?

98. Что такое указатель на *void*? Зачем нужны такие указатели?

99. Какие операции допустимы при работе с указателями?

100. Чем отличается унарная операция "&" от унарной "*" ?

101. Совместимость типов указателей.

102. Можно ли получить адрес указателя?

103. Можно ли указателю присвоить его же адрес?

104. Почему к указателю на *void* нельзя применить операцию разыменования?

105. Как работают операции инкремента и декремента, примененные к указателям?

106. Каков результат операции вычитания, примененной к указателям одного типа?

107. Какой спецификатор типа используется при выводе адреса на экран с помощью функции *printf()*?

108. В чем отличие записи *(double *) a* от *(double) * a*, если *a* – указатель на целое число?

109. В чем отличие записи **a++* от *(*a)++*, если *a* – некоторый указатель, отличный от *void**?

110. Как описать указатель на начало массива?

111. Как описать указатель на указатель?

112. Когда и зачем может повторно использоваться операция разыменования?

113. Что такое массив?

114. Что представляет собой имя массива?

115. Что представляет собой индекс элемента массива?

116. Как можно обратиться к элементу массива?

117. Как получить адрес элемента массива?

118. Как описать указатель на начало массива?
119. Как обратиться к элементу массива через указатель?
120. Существует ли связь между индексом элемента и его значением?
121. Чему равен индекс первого элемента массива?
122. Как проинициализировать массив?
123. Когда можно не указывать количество элементов массива при описании?
124. Какого типа могут быть элементы массива?
125. Какие операции можно производить над целым массивом?
126. Почему при обработке массивов используют циклы?
127. Как поменять местами два элемента массива?
128. Чему равен индекс последнего элемента массива?
129. Возникнет ли ошибка при обращении к элементу массива, индекс которого больше индекса последнего элемента этого массива?
130. Может ли существовать массив из одного элемента? Если может, то как его описать?
131. Является ли p указателем на массив $a[]$, если они объявлены следующим образом: `int a[10], *p=a;` ?
132. Что такое сортировка массива?
133. Что такое двумерный массив?
134. Как располагается двумерный массив в памяти компьютера?
135. Как проинициализировать двумерный массив?
136. Почему при объявлении двумерного массива с одновременной инициализацией первые скобки можно оставить пустыми, а вторые нет?
137. При обращении к элементу двумерного массива сначала указывается индекс строки или индекс столбца?
138. Как вывести двумерный массив таблицей с колонками одинаковой ширины?
139. Что представляет собой имя двумерного массива?
140. Что представляет собой первый индекс элемента двумерного массива?
141. Что представляет собой второй индекс элемента двумерного массива?
142. Что представляет собой имя строки матрицы?
143. Как получить адрес элемента двумерного массива?
144. Как получить адрес начала двумерного массива?
145. Как получить адрес строки матрицы?
146. Можно ли просмотреть все элементы двумерного массива в одном цикле?
147. Как обратиться к некоторому элементу двумерного массива через указатель на первый элемент матрицы?
148. Как обратиться к элементу двумерного массива через указатель на начало массива?
149. Как описать трехмерный массив и как с ним работать?
150. Какими способами можно запрограммировать матрицу?
151. Чем отличается просмотр элементов матрицы по столбцам от

просмотра по строкам?

152. Какова связь между индексами строки и столбца у элементов, лежащих на главной диагонали квадратной матрицы?

153. Какова связь между индексами строки и столбца у элементов, лежащих на побочной диагонали квадратной матрицы?

154. Можно ли просмотреть все элементы матрицы в одном цикле?

155. В каких областях оперативной памяти могут располагаться данные при выполнении программы?

156. Какие переменные называют динамическими? Чем они отличаются от статических переменных?

157. С какой целью используют динамические переменные?

158. Как обращаются к динамическим переменным?

159. Какие бывают указатели? Как их описать?

160. Каким образом можно выделять память для динамических переменных и освобождать её?

161. Как определить, выделена память или нет?

162. Почему нельзя забывать освобождать выделенную память?

163. Чему равно значение указателя после освобождения области памяти, на которую он указывал?

164. Как разместить в динамической памяти массив?

165. Какими способами можно разместить в динамической памяти матрицу?

Семестр 2

1. Что такое функция?

2. Что такое функция типа void?

3. Что такое прототип функции?

4. Чем отличается прототип функции от вызова функции?

5. Чем отличается описание функции от определения функции?

6. Критерии соответствия формальных и фактических параметров.

7. Соответствие типов формальных и фактических параметров.

8. Зачем нужен оператор return?

9. Может ли в функции быть несколько операторов return?

10. Когда необходимо писать оператор return в функции типа void?

11. Что такое побочный эффект функции?

12. Когда используются формальные параметры-указатели? Какими в этом случае должны быть фактические параметры?

13. Как передать массив в функцию? Как передать в функцию матрицу?

14. Как описать функцию, позволяющую работать и с одномерными массивами, и с матрицами?

15. Что такое параметр-константа? Когда используются такие параметры?

16. Что такое указатель на функцию? Как его описать?

17. Как передать функцию в функцию? Что будет являться формальным параметром, что фактическим? Как обратиться к функции, переданной

- через параметры?
18. Что такое рекурсия? Когда она применяется? Что такое глубина рекурсии?
 19. Как описать функцию с переменным числом параметров? Как использовать такую функцию?
 20. Какие параметры могут быть у функции `main()`? Как отлаживать и тестировать программу, содержащую функцию `main()` с параметрами?
 21. Способы организации строк.
 22. Что такое Си-строка? Чем она отличается от массива символов?
 23. Что такое нуль-терминатор?
 24. Чем отличается описание `char *st` от `char st[N]`, где N – некоторая константа?
 25. Как инициализировать строку?
 26. Чем отличается ввод строки с помощью функции `scanf()` от ввода с помощью функции `gets()`?
 27. Можно ли использовать операцию присваивания для задания значения строки?
 28. Как изменить строку?
 29. Что такое конкатенация строк?
 30. Какие функции для работы со строками Вы можете назвать?

Семестр 3

1. Структура программы на языке программирования C#.
2. Классификация типов данных. Встроенные типы данных. Преобразование типов.
3. Основные операторы. Чтение данных с клавиатуры. Вывод данных на экран.
4. Понятие исключительной ситуации.
5. Массивы. Размерность массива. Методы и свойства класса `Array`.
6. Многомерные массивы. Рваные массивы.
7. Регулярные и буквальные строковые литералы. Ссылки типа `string`. Операции над строками.
8. Методы и свойства класса `String`. Форматирование строк. Неизменяемость объектов класса `String`. Тип `System.Text.StringBuilder`.
9. Функции. Параметры функции. Функции с переменным числом параметров.
10. Функции. Функции с необязательными параметрами. Функции с именованными параметрами. Перегрузка методов.
11. Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
12. Потоки байтов `FileStream`.
13. Потоки символов `StreamWriter` и `StreamReader`.

14. Двоичные файлы BinaryWriter, BinaryReader.
15. Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора.
16. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы.
17. Перегрузка операций (унарные и бинарные операции, приведение типов).
18. Основные свойства ООП. Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.
19. Основные свойства ООП. Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания.
20. Абстрактные классы. Класс object. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET
21. Понятие делегата. Описание и использование. Многоадресная передача.
22. События. События для многоадресной передачи.
23. Механизм обработки исключений. Системные исключения и их обработка. Свойства и методы класса Exception. Исключения, создаваемые программистом.
24. Основные абстрактные структуры данных. Физическое представление данных.
25. Пространство имен System.Collections. Интерфейсы пространства имен System.Collections.
26. Коллекции общего назначения. Основные элементы класса ArrayList.
27. Коллекции общего назначения. Основные элементы класса Hashtable.
28. Коллекции общего назначения. Основные элементы класса SortedList.
29. Коллекции общего назначения. Основные элементы класса Stack.
30. Коллекции общего назначения. Основные элементы класса Queue.
31. Коллекции общего назначения. Доступ к коллекциям с помощью нумератора.
32. Классы-прототипы. Основные преимущества использования обобщений. Ограничения.
33. Параметризованные коллекции библиотеки .NET.
34. Создание собственных классов-прототипов. Обобщенные методы.

Критерии и шкала оценивания по оценочному средству фронтальные и индивидуальные опросы

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Ответы на контрольные вопросы выполнены на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Ответы на контрольные вопросы выполнены на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Ответы на контрольные вопросы выполнены на низком уровне

	(правильные ответы даны на 50-74% вопросов/задач)
2	Ответы на контрольные вопросы выполнены на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Контрольные работы

Типовые варианты контрольных работ

Семестр 1

1. Вычислить количество точек с целочисленными координатами, попадающих в круг радиуса R с центром в точке с координатами X, Y . Учитывать только те точки, координата X которых четная, а координата Y нечетная.
2. Определить, из скольких различных букв состоит текст, содержащий N строк. Текст представить в виде массива.
3. Найти минимум и максимум функции $f(x, n) = \sum_{i=1}^n \left(i^x + e^{\frac{i+x}{n}} \right)$ на отрезке $[A, B]$, задаваемом пользователем. Значение n и количество шагов также задается пользователем.

Семестр 2

1. Найти все точки пересечения графиков функций $f_1(x, n) = \sum_{i=1}^n (i^x + x^i)$ и $f_2(x, n) = \sum_{i=1}^n \left(\frac{x+i}{n} \right)^{\sin(x)}$ на заданном интервале $[A, B]$. (Графики считаются пересекающимися, если $|f_1(x) - f_2(x)|$ меньше заданного значения E . Количество шагов по оси X задается пользователем. Вычисление каждой функции оформить подпрограммой.
2. Выяснить, имеется ли в данной прямоугольной матрице A элемент, такой, что он является одновременно максимальным в строке и минимальным в столбце, на пересечении которых он находится. Если такой элемент существует, то напечатать его значение и индексы. Поиск максимального элемента строки и минимального элемента столбца оформить как пользовательскую функцию.

3. Используя рекурсивную функцию, вычислить значение выражения

$$f(n) = \sum_{i=1}^n i!$$

Семестр 3

1. Разработать класс «набор вещественных чисел», который имеет методы: поиск минимума, поиск максимума, сумма чисел внутри заданного диапазона
2. Разработать иерархию классов для отображения графических фигур

Критерии и шкала оценивания по оценочному средству контрольные работы

Шкала оценивания (интервал баллов)	Критерий оценивания
5	Ответы на контрольные вопросы выполнены на высоком уровне (правильные ответы даны на 90-100% вопросов/задач)
4	Ответы на контрольные вопросы выполнены на среднем уровне (правильные ответы даны на 75-89% вопросов/задач)
3	Ответы на контрольные вопросы выполнены на низком уровне (правильные ответы даны на 50-74% вопросов/задач)
2	Ответы на контрольные вопросы выполнены на неудовлетворительном уровне (правильные ответы даны менее чем на 50%)

Оценочные средства для промежуточной аттестации (экзамен)

Типовой билет. Семестр I

ЭКЗАМЕН

по дисциплине: «Программное обеспечение мехатронных и робототехнических систем».

Первый семестр.

Направление подготовки: 15.03.06 Мехатроника и робототехника

Профиль подготовки: Компьютерные технологии управления в робототехнике и мехатронике

БИЛЕТ ПЕРВЫЙ

- 1 Написать программу, проверяющую, можно ли построить треугольник из отрезков с длинами **A,B,C**. (отрезки имеют целочисленную длину). Если это возможно, то определить, является ли этот треугольник прямоугольным.
- 2 В массиве из 1000 целых положительных чисел, не превосходящих по значению десяти, определить, сколько чисел равно единице, сколько двойке и т.д. Напечатать результат.
- 3 Переставить элементы квадратной матрицы размером 20 на 20 симметрично главной диагонали.

Билет рассмотрен и утвержден на заседании кафедры «**Информационные и управляющие системы**» от «__» _____ 2023 г. (протокол № __)

Заведующий кафедрой

Горбунов А.И.

Преподаватель

Юрков Д. А.

ЭКЗАМЕН

по дисциплине: «Программное обеспечение мехатронных и робототехнических систем».

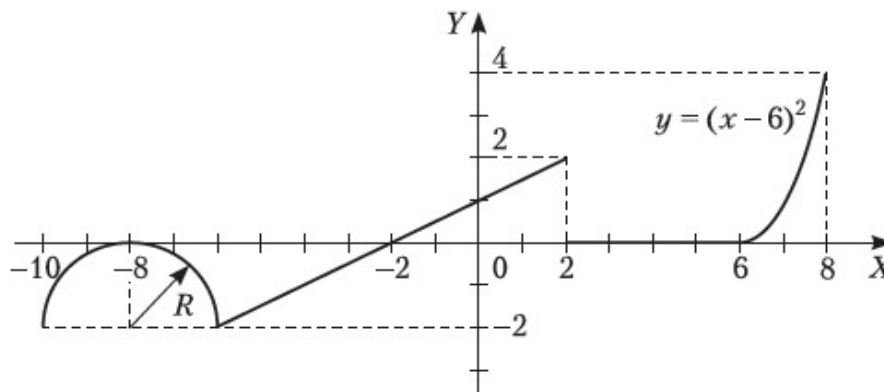
Второй семестр.

Направление подготовки: 15.03.06 Мехатроника и робототехника

Профиль подготовки: Компьютерные технологии управления в робототехнике и мехатронике

БИЛЕТ ПЕРВЫЙ

- 1 Написать программу, которая по введенному значению аргумента x вычисляет и выводит на экран значение функции y , заданной в виде графика. Параметр R задается константой.



- 2 Дан массив целого типа размером $M \times N$ элементов (числа M и N вводятся с клавиатуры). Написать функцию, которая имеет следующие параметры: данный массив, его размерности, и возвращает структуру, содержащую сумму четных элементов в нечетных столбцах массива, а также количество этих элементов и столбцов. Вывести на экран информацию, содержащуюся в возвращаемой структуре и обеспечить корректное освобождение выделяемых ресурсов в основной программе.
- 3 Написать программу, которая определяет, принадлежит ли точка с введенными координатами (x, y) функции y , заданной в виде графика (Задача 1). Результат работы программы вывести на экран в виде текстового сообщения (да/нет). Решить задачу, используя рекурсивную функцию.

Билет рассмотрен и утвержден на заседании кафедры «**Информационные и управляющие системы**» от «___» _____ 2023 г. (протокол № ___)

Заведующий кафедрой

Горбунов А.И.

Преподаватель

Юрков Д. А.

Типовой билет. Семестр 3

ЭКЗАМЕН

по дисциплине: «Программное обеспечение мехатронных и робототехнических систем».

Третий семестр.

Направление подготовки: 15.03.06 Мехатроника и робототехника

Профиль подготовки: Компьютерные технологии управления в робототехнике и мехатронике

БИЛЕТ ПЕРВЫЙ

- 1

 Дайте определение следующих понятий: абстракция, инкапсуляция, наследование и полиморфизм
- 2

 Реализовать класс «стек» с поддержкой операций push (добавить элемент), pop (взять элемент), и count (узнать количество элементов). Элементы стека могут быть разных типов.
- 3

 Создать класс «Матрица». Класс должен иметь следующие свойства:
 1. двумерный массив вещественных чисел;
 2. количество строк и столбцов в матрице.

Класс должен иметь следующие методы:

1. сложение с другой матрицей;
 2. умножение на число;
 3. умножение матриц.
-

Билет рассмотрен и утвержден на заседании кафедры «**Информационные и управляющие системы**» от «__» _____ 2023 г. (протокол № __)

Заведующий кафедрой

Горбунов А.И.

Преподаватель

Юрков Д. А.

Критерии и шкала оценивания по оценочному средству промежуточный контроль (экзамен)

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

Оценочные средства для промежуточной аттестации (курсовой проект)

По объему курсовая работа состоит из пояснительной записки объемом от 20 до 30 страниц компьютерного текста и (при необходимости) презентацию до 7 слайдов. Имеет следующую рекомендуемую структуру (разделы):

1. **Титульный лист.**
Задание на курсовую работу, в котором отражается содержание выполняемой работы.
2. **Введение**, в котором излагаются основные цели и задачи проектирования.
3. **Теоретическая часть**, в которой изложены основные идеи и технологии, используемые при разработке программного обеспечения. Приводится описание алгоритмов функционирования программного обеспечения, обосновываются задачи проектирования, проводится анализ известных решений на основе анализа технической и научной литературы.
4. **Практическая часть**, в которой излагаются технические решения, использованные при разработке программного обеспечения. Приводится подробное описание и общие принципы построения проектируемой системы, отдельных элементов (блоков) системы, дается описание функциональных характеристик, обосновывается выбор аппаратных средств, пригодных для эксплуатации программного обеспечения. Приводятся примеры функционирования программного обеспечения.
5. **Заключение**, в котором приводятся основные результаты выполненной работы, даются рекомендации по использованию принятых решений.
6. Список использованной литературы.
7. **Приложения**, в которые включаются дополнительные справочные материалы, листинги программ.

Типовые задания к курсовому проекту

1. Разработать программу - просмотрщик рисунков (изображений), многофункциональный. Программа должна позволять открывать и просматривать изображения, сохранять их под другим именем. При необходимости пользователь может повернуть изображение на 90^0 . Организовать возможность просмотра рисунков в директории в режиме «Slide Show» (показ слайдов). Должна быть организована возможность просмотра изображения в полном экране с оптимизацией размера просматриваемого изображения по высоте или по ширине. В строке состояния должны показываться сведения об изображении.
2. Написать программу «Лабиринт» (пример главного окна игры показан на рисунке 1) в графическом режиме. Программа должна генерировать

лабиринт (либо загружать из графического файла), а игрок, управляя объектом, должен вывести этот объект из лабиринта.

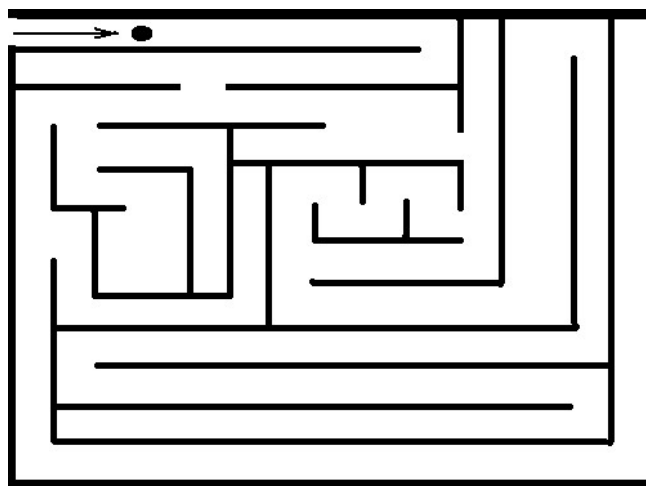


Рисунок 1. – Возможный вид программы для игры «Лабиринт»

3. Стрельба по мишеням» (пример главного окна игры показан на рисунке 2) в графическом режиме. В качестве прицела использовать пересечение вертикальной и горизонтальной линий, управляемых клавишами со стрелками. Мишени различного размера двигаются по экрану с разной скоростью и в различном направлении.

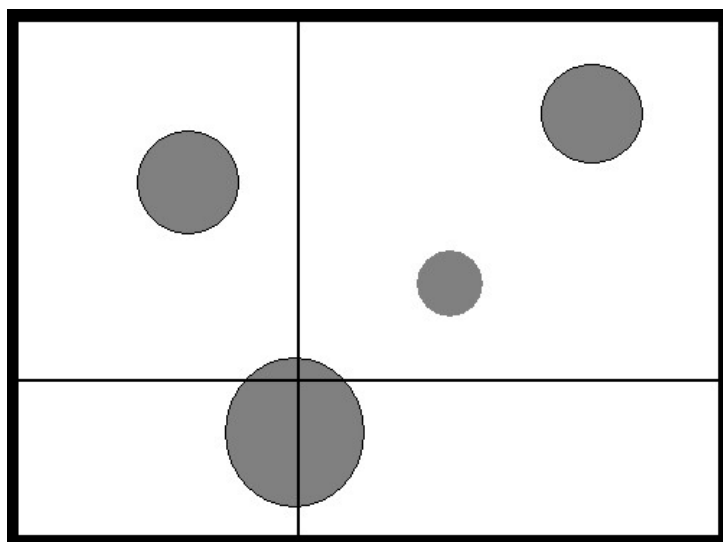


Рисунок 2. – Возможный вид игры «Стрельба по мишеням»

4. Написать игру «Арканоид» (пример главного окна игры показан на рисунке 3) в графическом режиме. Суть игры: имеется поле, в верхней части которого располагаются ряды прямоугольных блоков, а в нижней части – бита, управляемая пользователем (движение вправо и влево); необходимо «выбить» все блоки мячом, отбивая его битой.

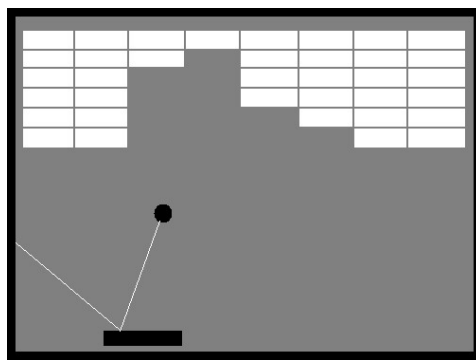


Рисунок 4. – Возможный вид игры «Арканойд»

5. Написать свою версию игры «Тетрис» (пример главного окна игры показан на рисунке 5).

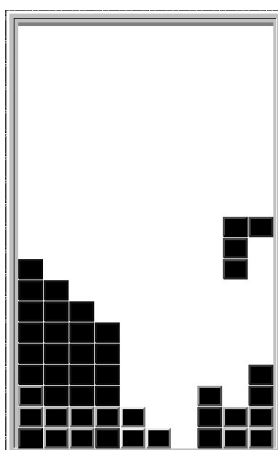


Рисунок 5. – Возможный вид игры «Тетрис»

Кроме приведенных примеров, возможна разработка таких приложений, как:

1. Система визуального отображения движения мобильного колёсного робота.
2. Система визуального отображения движения мобильного биоподобного робота.
3. Система визуального отображения движения манипулятора.
4. Система визуального отображения движения антрополоподобного робота.
5. Система визуального отображения функционирования простой механической системы.
6. Система визуального отображения летающего автономного робота.
7. Игровые задачи
8. Визуализация процесса имитационного моделирования группы автономных роботов
9. Визуализация процесса имитационного моделирования производственного процесса механообработки
10. Визуализация процесса имитационного моделирования системы управления

Критерии и шкала оценивания по оценочному средству курсовой проект

Шкала оценивания (интервал баллов)	Критерий оценивания
отлично (5)	Студент глубоко и в полном объеме владеет программным материалом. Грамотно, исчерпывающе и логично его излагает в устной или письменной форме. При этом знает рекомендованную литературу, проявляет творческий подход в ответах на вопросы и правильно обосновывает принятые решения, хорошо владеет умениями и навыками при выполнении практических задач.
хорошо (4)	Студент знает программный материал, грамотно и по сути излагает его в устной или письменной форме, допуская незначительные неточности в утверждениях, трактовках, определениях и категориях или незначительное количество ошибок. При этом владеет необходимыми умениями и навыками при выполнении практических задач.
удовлетворительно (3)	Студент знает только основной программный материал, допускает неточности, недостаточно четкие формулировки, непоследовательность в ответах, излагаемых в устной или письменной форме. При этом недостаточно владеет умениями и навыками при выполнении практических задач. Допускает до 30% ошибок в излагаемых ответах.
неудовлетворительно (2)	Студент не знает значительной части программного материала. При этом допускает принципиальные ошибки в доказательствах, в трактовке понятий и категорий, проявляет низкую культуру знаний, не владеет основными умениями и навыками при выполнении практических задач. Студент отказывается от ответов на дополнительные вопросы.

Форма листа изменений и дополнений, внесенных в ФОС

Лист изменений и дополнений

№ п/п	Виды дополнений и изменений	Дата и номер протокола заседания кафедры (кафедр), на котором были рассмотрены и одобрены изменения и дополнения	Подпись (с расшифровкой) заведующего кафедрой (заведующих кафедрами)

Экспертное заключение

Представленный фонд оценочных средств (далее – ФОС) по дисциплине «Программное обеспечение мехатронных и робототехнических систем» соответствует требованиям ФГОС ВО.

Предлагаемые формы и средства текущего и промежуточного контроля адекватны целям и задачам реализации основной профессиональной образовательной программы по направлению подготовки 15.03.06 Мехатроника и робототехника.

Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины представлены в полном объеме.

Виды оценочных средств, включенные в представленный фонд, отвечают основным принципам формирования ФОС.

Разработанный и представленный для экспертизы фонд оценочных средств рекомендуется к использованию в процессе подготовки обучающихся по указанному направлению.

Председатель учебно-методической
комиссии факультета компьютерных
систем и информационных
технологий



Ветрова Н. Н.