

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)**

Институт информационных технологий и радиоэлектроники



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
ИНФОРМАЦИОННО-ПОИСКОВЫЕ ЯЗЫКИ

направление подготовки
09.03.01 Информатика и вычислительная техника

направленность (профиль) подготовки
Системы автоматизированного проектирования микроэлектроники

г. Владимир

2021

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью освоения дисциплины «Информационно-поисковые языки» является формирование у студентов знаний по применению средств объектно-ориентированного программирования вместе с реализацией на достаточно низком уровне.

Задачи: - Обучение студентов основам алгоритмизации вычислительных процессов и общим принципам программирования, представления основных структур программ и данных.

- Создание фундаментальной основы знаний, необходимой при проектировании программ для вычислительных систем.

- Изучение основных этапов решения задач на ЭВМ.

- Изучение синтаксических и семантических конструкций языков программирования высокого уровня.

- Освоение студентами основных приемов программирования типовых задач.

- Изучение особенностей реализации обработки различных типов данных.

- Получение навыков программирования на языке C++.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Информационно-поисковые языки» относится к части, формируемая участниками образовательных отношений.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОПОП (компетенциями и индикаторами достижения компетенций)

Формируемые компетенции (код, содержание компетенции)	Планируемые результаты обучения по дисциплине, в соответствии с индикатором достижения компетенции		Наименование оценочного средства
	Индикатор достижения компетенции (код, содержание индикатора)	Результаты обучения по дисциплине	
ПК-3 Способен проводить научно-исследовательские и опытно-конструкторские разработки по отдельным разделам темы	ПК-3.1 Знает инструментальный математического анализа дискретных объектов и систем ПК 3.2 Умеет анализировать и формализовать полученные на практике или при исследованиях результаты и делать на их основе обоснованные выводы ПК 3.3 Владеет навыками применения методов решения теоретических задач в области схемотехники цифровых устройств	<i>Знать:</i> основные тенденции в области развития технологий объектно-ориентированного программирования. <i>Уметь:</i> ставить задачу и разрабатывать алгоритм ее решения на современных ЭВМ, использовать современные методы объектно-ориентированного программирования при кодировании	Тестовые вопросы Практико-ориентированное задание

		программных систем разного уровня сложности. <i>Владеть:</i> методами проектирования объектно- ориентированных программ, навыками работы с технической и справочной литературой.	
--	--	--	--

4. ОБЪЕМ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 7 зачетных единиц, 252 часов

Тематический план форма обучения – очная

№ п/п	Наименование тем и/или разделов/тем дисциплины	Семестр	Неделя семестра	Контактная работа обучающихся с педагогическим работником				Самостоятельная работа	Формы текущего контроля успеваемости, форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные работы	в форме практической подготовки		
1	Основы алгоритмического языка C++	3	1-2	2	2	4	2	14	
2	Объекты	3	3-4	1	1	2	1	14	
3	Состояние, поведение и индивидуальность объектов	3	4-5	2	1	4	1	14	
4	Классы	3	5-6	2	2	2	1	15	Рейтинг контроль 1
5	Отношения между объектами	3	7-8	1	2	4	1	14	
6	Простое наследование классов	3	9-10	2	2	4	1	14	
7	Множественное наследование классов	3	11-12	2	2	4	2	15	Рейтинг контроль 2
8	Средства повышения надежности и уровня абстракции программирования	3	13-14	2	2	4	1	14	
9	Обобщенные классы	3	15-16	2	2	4	1	15	
10	Перегрузка операторов	3	17-18	2	2	4	1	15	Рейтинг контроль 3
Всего за 3 семестр:				18	18	36		144	Экзамен
Наличие в дисциплине КП/КР									Курсовая работа
Итого по дисциплине				18	18	36		144	Экзамен, Курсовая работа

Содержание лекционных занятий по дисциплине

Раздел 1. Основы алгоритмического языка C++

Тема 1. Основы алгоритмического языка C++

Примеры простейших программ. Типы данных. Динамическое распределение памяти. Дополнительные операторы присваивания. Инструкции. Функции.

Раздел 2. Объекты

Тема 1. Объекты

Рассмотрим определения понятия объект с разных точек зрения. Объявление классов и создание объектов в программе на C++.

Раздел 3. Состояние, поведение и индивидуальность объектов

Тема 1. Состояние, поведение и индивидуальность объектов

Определения понятий состояние, поведение и индивидуальность объектов. Примеры программ.

Раздел 4. Классы

Тема 1. Классы

Определения понятия класс. Обозначение класса на языке UML.

Раздел 5. Отношения между объектами

Тема 1. Отношения между объектами

Виды отношений между объектами. Наследование. Отношение ассоциации. Видимость на уровне классов и на уровне объектов. Отношение реализации. Отношение дружественности. Отношение наполнения.

Раздел 6. Простое наследование классов

Тема 1. Простое наследование классов

Обозначение наследования в UML. Добавление полей и методов. Переопределение и уточнение методов. Таблица виртуальных методов. Особенности работы с конструкторами и деструкторами. Запрет наследования. Различие понятий «подкласс» и «подтип».

Раздел 7. Множественное наследование классов

Тема 1. Множественное наследование классов

Примеры программ. Неопределенность в наименовании. Повторное наследование.

Раздел 8. Средства повышения надежности и уровня абстракции программирования

Тема 1. Средства повышения надежности и уровня абстракции программирования

Потоковый ввод/вывод в C++. Перегрузка операторов потокового ввода и вывода. Работа с файлами последовательного доступа. Константные методы, константные поля и объекты-константы. Статические методы и статические поля класса. Конструктор копирования и конструктор преобразования. Функции преобразования типов. Обработка исключений. Пространства имен.

Раздел 9. Обобщенные классы

Тема 1. Обобщенные классы

Отношение наполнения. Специализация методов обобщенного класса. Эквивалентность типов. Отношения между обобщенными классами.

Раздел 10. Перегрузка операторов

Тема 1. Перегрузка операторов

Ограничения на перегрузку операторов. Примеры программ. Перегрузки операторов для типов, не являющихся классом. Функция-оператор как утилита, как метод какого-либо класса.

Содержание лабораторных занятий по дисциплине

Раздел 1. Основы алгоритмического языка C++

Лабораторная работа № 1. Изучение основ графического интерфейса GDI в Microsoft Windows. Получить представление о работе с ним с использованием алгоритмического языка C++.

Раздел 2-4. Объекты. Состояние, поведение и индивидуальность объектов Классы.

Лабораторная работа № 2. Изучение основ объектно-ориентированного программирования на алгоритмическом языке C++.

Раздел 5. Отношения между объектами

Лабораторная работа № 3. Изучение основ объектно-ориентированного программирования на алгоритмическом языке C++. Получить представление об организации инкапсуляции, наследования и полиморфизма.

Раздел 6. Простое наследование классов

Лабораторная работа № 4. Изучение основ множественного наследования классов в алгоритмическом языке C++. Исследовать особенности выполнения конструкторов и деструкторов при использовании виртуальных базовых классов.

Раздел 7. Множественное наследование классов

Лабораторная работа № 5. Изучение средств реализации ассоциаций между классами в алгоритмическом языке C++.

Раздел 8. Средства повышения надежности и уровня абстракции программирования

Лабораторная работа № 6. Изучение средств обобщения классов в языке C++.

Раздел 9. Обобщенные классы

Лабораторная работа № 7. Изучение механизма перегрузки операторов в языке C++.

Раздел 10. Перегрузка операторов

Лабораторная работа № 8. Изучение механизма статических методов и статических полей класса в языке C++.

Содержание практических занятий по дисциплине

Раздел 1. Основы алгоритмического языка C++

Практическая работа 1. Основы алгоритмического языка C++

Раздел 2-3. Объекты. Состояние, поведение и индивидуальность объектов

Практическая работа 2. Основы определения объектов в предметной области.

Раздел 4. Классы

Практическая работа 3. Основы создания диаграмм UML.

Раздел 5. Отношения между объектами

Практическая работа 4. Отношения между объектами.

Раздел 6. Простое наследование классов

Практическая работа 5. Простое наследование классов.

Раздел 7. Множественное наследование классов

Практическая работа 6. Множественное наследование классов

Раздел 8. Средства повышения надежности и уровня абстракции программирования

Практическая работа 7. Средства повышения надежности и уровня абстракции программирования

Раздел 9. Обобщенные классы

Практическая работа 8. Обобщенные классы.

Раздел 10. Перегрузка операторов

Практическая работа 9. Перегрузка операторов.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

5.1. Текущий контроль успеваемости

Рейтинг-контроль №1

Вопросы

1. Директивы препроцессора языка C++.
2. Встроенные типы данных в языке C++.
3. Константы. Инициализация переменных.
4. Константные типы.
 1. Перечисляемые типы.
 2. Синонимы типов.

3. Арифметические операторы,
4. Логические операторы,
5. Структуры.
6. Типизированные указатели.
7. Ссылочный тип.
8. Организация одномерных и многомерных массивов.
9. Составные операторы присваивания.
10. Операторы инкремента и декремента.
11. Оператор ветвления.
12. Оператор выбора.
13. Оператор цикла с предусловием.
14. Оператор цикла с постусловием.
15. Оператор итеративного цикла.
16. Операторы безусловного перехода.
17. Функции в C++. Способы передачи параметров.
18. Перегрузка функций.
19. Прототип функции.
20. Шаблоны функций.
21. Рекурсия.
22. Понятие «объект» с различных точек зрения

Примечание: В РК-1 требуется решить матричную задачу, предусматривающую описание и использование функций, заполнение матрицы 2-мя способами (ручной ввод и генерация случайных чисел), расчет (выбор) требуемых по условию задачи величин, вывод матрицы и вектора и динамическое распределение памяти для матрицы.

Рейтинг-контроль №2

Вопросы

1. . Объявление классов
2. . Создание объектов в программе на C++
3. . Состояние объекта.
4. . Назначения конструктора
5. . Способы создания объекта
6. . Поведение объекта.
7. . Индивидуальность объекта.
8. . Отношение использования между объектами
9. . Отношение наследования между классами
10. Конструктор копирования в C++.
11. Перегрузка операторов для типов, не являющихся классами.
12. Перегрузка операторов для классов.
13. Поточковый ввод
14. Поточковый вывод
15. Ошибки потоков.
16. Перегрузка операторов потокового ввода
17. Перегрузка операторов потокового вывода.

Примечание: В РК-2 требуется решить матричную задачу, предусматривающую описание и использование классов `Vector` и `Matrix` (матрица). Классы должны включать требуемые поля и методы: конструкторы, деструктор, заполнение матрицы 2-мя способами (ручной ввод и генерация случайных чисел), расчет (выбор) требуемых по условию задачи величин, вывод матрицы и вектора и динамическое распределение памяти для объекта класса `Matrix`.

Рейтинг-контроль №3

Вопросы

1. . Неопределенность в наименовании при множественном наследовании классов.
2. . Проблема повторного наследования при множественном наследовании классов. Виртуальные базовые классы.
3. Дружественные классы и функции.
4. Шаблоны классов.
5. Агрегирование объектов, инициализаторы полей.
6. Множественное наследование классов.
7. Оператор `this`.
8. .Работа с файлами последовательного доступа в C++ средствами потокового ввода/вывода. Виды ввода/вывода.
9. Набор классов и объектов для файлового ввода/вывода.
10. Форматированный вывод в файл.
11. Форматированное чтение из файла.
12. Посимвольный форматированный вывод и ввод.
13. Двоичный ввод/вывод числовых данных.
14. Объектный ввод/вывод. Сохранение объекта в файле.
15. Объектный ввод/вывод. Чтение объекта из файла.
16. Объектный ввод/вывод. Чтение и запись нескольких объектов.
17. . Константные методы и константные поля.
18. . Статические методы и статические поля класса.

Примечание: В РК-3 требуется решить задачу, предусматривающую описание и использование классов и их наследование, включающую требуемые поля и методы: конструкторы, деструктор. В задаче необходимо создать файл, содержащий экземпляры класса, и выполнить поиск экземпляров, удовлетворяющих заданным требованиям.

5.2. Промежуточная аттестация по итогам освоения дисциплины

Перечень вопросов к экзамену

1. Директивы препроцессора C++.
2. Встроенные типы данных в C++.
3. Константы. Инициализация переменных.
4. Константные типы.
5. Перечисляемые типы.
6. Синонимы типов.
7. Арифметические операторы,
8. Логические операторы,
9. Структуры.
10. Типизированные указатели.
11. Ссылочный тип.
12. Организация одномерных и многомерных массивов.
13. Составные операторы присваивания.
14. Операторы инкремента и декремента.
15. Оператор ветвления.
16. Оператор выбора.
17. Оператор цикла с предусловием.
18. Оператор цикла с постусловием.
19. Оператор итеративного цикла.
20. Операторы безусловного перехода.
21. Функции в C++. Способы передачи параметров.

22. Перегрузка функций.
23. Прототип функции.
24. Шаблоны функций.
25. Понятие «объект» с различных точек зрения
26. Объявление классов
27. Создание объектов в программе на C++
28. Состояние объекта.
29. Назначения конструктора
30. Способы создания объекта.
31. Поведение объекта.
32. Индивидуальность объекта.
33. Отношение использования между объектами
34. Отношение включения между объектами
35. Отношение наследования между классами
36. Переопределение методов – виды.
37. Таблица виртуальных методов
38. Особенности работы с конструкторами и деструкторами при наследовании.
39. Уточнение методов.
40. Формы наследования.
41. Отношение использования между классами
42. Конструктор копирования в C++.
43. Перегрузка операторов для типов, не являющихся классами.
44. Перегрузка операторов для классов.
45. Поточковый ввод
46. Поточковый вывод
47. Перегрузка операторов потокового ввода
48. Перегрузка операторов потокового вывода.
49. Множественное наследование классов.
50. Неопределенность в наименовании при множественном наследовании классов.
51. Проблема повторного наследования при множественном наследовании классов.
Виртуальные базовые классы.
52. Дружественные классы и функции.
53. Шаблоны классов.
54. Агрегирование объектов, инициализаторы полей.
55. Работа с файлами последовательного доступа в C++ средствами потокового ввода/вывода.
56. Константные методы и константные поля.
57. Статические методы и статические поля класса.
58. Связь основных этапов жизненного цикла программного средства.
59. Этапы проектирования (создания) программного средства.
60. Проблемы при разработке сложных программных средств.
61. Стратегии проектирования.
62. Основные диаграммы (Д) при объектно-ориентированном подходе:
 - диаграммы классов и отношений наследования;
 - диаграммы отношений использования;
 - диаграммы объектов;

Примечание: В экзаменационном билете дается задача, предусматривающая использование шаблонов классов, параметризацию классов, перегрузку операторов и работу с файлами.

5.3. Самостоятельная работа обучающегося.

Целью самостоятельной работы являются формирование личности студента, развитие его способности к самообучению и повышению своего профессионального уровня.

Самостоятельная работа заключается в изучении содержания тем курса по конспектам, учебникам и дополнительной литературе, подготовке к лабораторным и практическим занятиям, выполнении курсовой работы, оформлении лабораторных работ и пояснительной записки к курсовой работе, подготовке к рубежным контролям, к экзамену.

Перечень тем для СРС

1. Свойства объектов.
2. Виды взаимодействия объектов.
3. Проблемы при множественном наследовании классов.
4. Параметризация обобщенного класса различными типами данных.
5. Параметризация обобщенного класса типами данных, введенных пользователем.
6. Диаграммы UML.

Темы курсовой работы.

Вариант 1. "Электронная шахматная доска"

Требуется создать программную систему для игры в шахматы двух игроков. Управление фигурами осуществляется с клавиатуры: курсором указывается фигура, подтверждается ее выбор, после чего указывается клетка, в которую эта фигура перемещается. Если в целевой клетке находится фигура противника, то она удаляется с поля. Фигуры одного цвета не могут уничтожать друг друга. Фигуры черного и белого цвета могут двигаться только по очереди, очередность хода показывается соответствующей надписью. Должна контролироваться правильность перемещения различных фигур (слон - по диагонали, ладья - по горизонтали и вертикали и т.д.).

Остальные тонкости шахматной игры (нападение на короля, рокировка и т.п.) в этом примере можно опустить. Ошибочные действия игроков сопровождаются текстовыми сообщениями.

Вариант 2. "Электронная доска для игры в шашки"

Требуется создать программную систему для игры в шашки двух игроков. Управление шашками осуществляется с клавиатуры: курсором указывается шашка, подтверждается ее выбор, после чего указывается клетка (в соответствии с правилами этой игры - по диагонали вперед), в которую эта шашка перемещается. Целевая клетка должна быть свободной. Реализовать "съедение" шашки противника, расположенной спереди и сзади. При достижении противоположной стороны доски шашка превращается в дамку с соответствующими правилами перемещения и взятия шашек противника. Взятие нескольких шашек противника подряд и другие тонкости игры можно опустить.

Вариант 3. "Прибор для исследования оптических приборов"

Это задание является упрощенным вариантом примера из [Буч]. Моделируется установка, состоящая из источника света, оптического стола (линейка с делениями, показывающими расстояние от источника) и набора из четырех линз (две - собирающие и две рассеивающие) с различными фокусными расстояниями. Функция модели - изображение пути лучей, проходящих через линзы, расположенные на оптической скамье. Пользователь должен иметь возможность выбора произвольного числа линз из имеющихся и установки их на оптической скамье в различных позициях. Кроме того источник света может быть перемещен в вертикальном направлении.

Вариант 4. Многооконный текстовый редактор

Результатом выполнения данного задания должен быть текстовый редактор, реализующий следующие функции: чтение текста из файла, запись текста в файл, переключение между окнами, удаление символов клавишами редактирования, перемещение курсора к

произвольной позиции на экране, выделение строчного блока, удаление строчного блока в "карман" и восстановление строчного блока из "кармана". Для упрощения реализации можно считать что размеры окна не превышают размеров экрана дисплея.

Вариант 5. Табличный процессор

Результатом выполнения данного задания должен быть табличный процессор, реализующий следующие функции: чтение таблицы из файла, запись таблицы в файл, перемещение курсора по клеткам таблицы, запись в клетку числа, запись в клетку текста, запись в клетку ссылки на другую клетку, запись в клетку функции суммирования блока клеток. При каждом изменении содержимого клеток вся таблица должна пересчитываться. Придерживаться схемы диалога и синтаксиса записей принятого в табличном процессоре Lotus 1-2-3. Для упрощения реализации можно считать, что размеры таблицы не превышают размеры экрана дисплея.

Вариант 6. Игра "Удав и кролики"

Прямоугольное игровое поле состоит из клеток. В пустых клетках случайным образом рождаются "кролики". "Удав" первоначально состоит из одной клетки (головы), расположенной в центре поля. При первом нажатии любой управляющей клавиши (управляющие клавиши - клавиши со стрелками) "удав" начинает двигаться в соответствующем направлении. При прохождении клетки с "кроликом" длина "удава" увеличивается на одну клетку. Нажатиями клавиш изменяется направление движения головы "удава". За пределами поля отображается количество "кроликов" на поле и длина "удава". При попадании "удава" в край игрового поля или в свой хвост игра останавливается.

Вариант 7. "Солнечная система"

Моделируется движение планет в солнечной системе и прохождение комет через нее. В упрощенной модели не следует соблюдать реальные пропорции размеров планет, Солнца и расстояний между ними. В данной модели все планеты движутся вокруг Солнца в одной плоскости но с разной скоростью. У планет, Солнца и комет есть определенные размеры и масса, пропорциональная кубу линейного размера. В одном из углов "Солнечной системы" в случайные моменты времени генерируются кометы, летящие в произвольном направлении. Комета и планета перестают существовать при столкновении. При столкновении кометы с Солнцем исчезает только комета. На кометы действуют силы притяжения планет и Солнца в соответствии с законом всемирного тяготения, что изменяет их траекторию. Комета вышедшая за пределы "Солнечной системы" исчезает навсегда. У комет есть хвост, направленный в сторону от Солнца. Управление моделью - при помощи клавиш: начать, закончить.

Вариант 8. Игра "Жизнь"

Компонентами клеточного автомата являются клетки (ячейки), образующие правильную двумерную решетку. Каждая клетка принимает значение из множества {"неживая", "живая"}. Значения всех клеток автомата одновременно обновляются при каждом "тикании часов" в соответствии со следующим правилом. Если у живой клетки имеется менее N_1 живых соседей, то она погибает от одиночества. Если у живой клетки более N_2 живых соседей, то она погибает от перенаселения. Если у неживой клетки от N_3 до N_4 живых соседей, то в ней появляется живая клетка. Живая клетка выделяется цветом в соответствии с количеством живых соседей. Последовательность выполнения следующая: в диалоге задаются N_1, N_2, N_3 ; при помощи курсора на пустое поле расставляются живые клетки; дается команда начала жизненного цикла; при необходимости процесс может быть приостановлен или выполнен по шагам. Один из вариантов параметров игры: $N_1=2, N_2=3, N_3=N_4=3$.

Вариант 9. Игра "Кубики-перевертыши"

Одинаковые кубики лежат в прямоугольной коробке. Каждый кубик окрашен в шесть цветов, по числу граней. Дно коробки разделено на квадраты. В каждом квадрате, кроме одного, лежит по кубику. За счет свободной ячейки кубики можно последовательно перекачивать из квадрата в квадрат. Вынимать и переворачивать кубики не разрешается. В исходном состоянии цветные грани кубиков располагаются одинаково. В головоломке требуется изменить цвет верхних граней кубиков на любой другой цвет, например, с белого на синий. При этом цвета боковых граней кубиков должны быть ориентированы одинаково. Пользователь должен иметь возможность указывать курсором перекачиваемый кубик, возвращать ходы на любую глубину, восстанавливать начальное состояние игры. Возникновение целевой ситуации должно сопровождаться соответствующим сообщением.

Вариант 10. "Тренажер теннисиста"

Тренажер представляет собой комнату с тремя стенами (в форме буквы П) у выхода которой находится теннисист. Теннисист может подавать мячи в произвольном направлении. Мячи отражаются от стен. Теннисист может отражать мячи двигаясь только поперек выхода. В процессе игры фиксируется число мячей, поданных теннисистом и число пропущенных мячей.

Вариант 11. Игра "Артиллерийская дуэль"

На поверхности, имеющей ломанный рельеф располагается два артиллерийских орудия. Пользователь может управлять направлением выстрела и начальной скоростью полета снаряда для каждого орудия. Траектория снаряда соответствует простейшим баллистическим соотношениям. Количество снарядов для каждого орудия ограничено. Игра заканчивается при попадании снаряда в одно из орудий, либо когда закончились все снаряды. Снаряды уничтожаются при столкновении между собой, либо при падении на земную поверхность. В цифровой форме фиксируется количество оставшихся снарядов для каждого орудия.

Вариант 12. Моделирование поведения стада животных

На территории леса в хаотичном порядке расположены деревья и животные. На одном из краев леса находится корм для животных. Во время движения на животных действуют два стимула: достижение корма и стремление быть близко друг к другу. Если животные оказываются ближе друг к другу определенного комфортного расстояния, то они стремятся разойтись. Если животное приближается к дереву, то у него возникает желание отодвинуться от препятствия. Необходимо выполнить имитацию движения стада к корму.

Вариант 13. Демографическая модель "Зайцы и волки"

На некоторой территории хаотически двигаются зайцы и волки, причем выйти за пределы этой территории они не могут. Если волк сталкивается с зайцем, то он его съедает. Заяц умирает от старости в заданном возрасте. За время жизни через равные промежутки времени каждый заяц рождает еще одного зайца. Волк умирает от старости со своей заданной продолжительностью жизни. Если волк в течение некоторого времени не съедает зайца, то он умирает от голода. Через равные промежутки времени волк рождает еще одного волка. В начальный момент времени зайцы и волки имеют случайным образом выбранный возраст. В процессе моделирования следует отображать в цифровой форме количество зайцев и их средний возраст, количество волков и их средний возраст. При инициализации процесса моделирования указываются параметры моделирования: начальное количество зайцев, начальное количество волков, продолжительность жизни зайцев, продолжительность жизни волков, интервал между рождением зайцев, интервал между рождением волков, интервал жизни волка после съедения последнего зайца.

* Вариант 14. Модель городского транспортного движения

Рассматривается перекресток двух дорог. Каждая дорога имеет две полосы движения в одну сторону и две - в другую. На перекрестке установлен светофор. Через равные промежутки времени с одной стороны светофора загорается зеленый свет, а с трех оставшихся сторон устанавливается красный свет. На дороге случайным образом заезжают транспортные средства двух видов: автомобили (с большой скоростью) и троллейбусы (с малой скоростью). Троллейбусы едут только по правой полосе. Автомобили могут двигаться по любой полосе. Если перед автомобилем оказывается троллейбус, то автомобиль переходит на левую сторону (совершает обгон). Если автомобиль движется по левой стороне, а справа нет троллейбуса, то автомобиль переходит на правую сторону. На перекрестке троллейбусы движутся только прямо, а автомобили с равной вероятностью поворачивают в любую сторону (но не назад). Транспортное средство не может двигаться, если перед ним красный свет светофора или другое транспортное средство, которое нельзя обогнать.

Вариант 15. "Восемь ферзей"

Имеется шахматная доска. Пользователь выставляет на нее восемь ферзей. Пользователю выдается сообщение о двух ситуациях: хотя бы один из ферзей оказывается под ударом других ферзей, ни один ферзь не находится под ударом других ферзей. Ферзи, находящиеся под ударом, и ферзи, не находящиеся под ударом, должны быть показаны различными цветами. При помощи курсора пользователь перемещает ферзей по полю. Два ферзя не могут быть расположены на одном поле. Должна быть предусмотрена возможность возврата ходов на произвольную глубину и их повторного выполнения.

Вариант 16. Прохождение конем шахматной доски

Реализуется следующая игра. Имеется шахматная доска. Курсором на доске указывается начальное положение коня. Далее курсором указывается следующее положение коня, в соответствии с правилами хода данной шахматной фигуры. Игра заключается в том, чтобы пройти всю шахматную доску, ни разу не попадая дважды в одну клетку. Если пользователь пытается выполнить ход не по правилам или поместить коня на клетку, которая уже была пройдена, то выдается предупредительное сообщение. Пользователю предоставляется возможность отменить любое количество предыдущих ходов. Если все клетки пройдены, то пользователю выдается поздравление. Если возникает ситуация, при которой не все клетки пройдены, а ход не может быть выполнен, то пользователю выдается соответствующее сообщение.

Вариант 17. Игра "Крестики-нолики"

Имеется поле, состоящее из квадратных клеток. Два игрока по очереди указывают клетку, в которую заносится либо крестик, либо нолик. Выигрывает тот игрок, который первым поставит пять своих знаков в ряд по вертикали, горизонтали или диагонали. Те знаки, которые образуют четверки и могут быть закончены выигрышем на следующем ходу, выделяются при помощи цвета. Предусмотреть возможность возврата игры на любое количество шагов назад. Пользователям сообщается, чей очередной ход должен быть сделан.

* Вариант 18. Игра "15"

Имеется квадратное игровое поле, состоящее из 16-ти клеток. На поле помещаются пятнадцать квадратных фишек с номерами от 1 до 15, т.е. одна клетка остается свободной. Пользователь передвигает фишки на освобождающуюся клетку так, чтобы в конце концов фишки были расположены по порядку номеров. Первоначальная установка фишек и их перемещение указываются пользователем при помощи курсора. При возникновении целевой ситуации пользователю выдается соответствующее сообщение. Пользователю предоставляется возможность возврата состояния игры на любое количество шагов назад.

Вариант 19. Графический редактор

Результатом выполнения данного задания должен быть упрощенный графический редактор со следующими функциями: выбор типа фигуры из заданного набора (отрезок, окружность, квадрат) и изображение соответствующей фигуры на рабочем поле в месте указанном курсором; выбор фигуры, расположенной на рабочем поле, и ее перемещение в другую позицию; удаление фигуры с рабочего поля; последовательное переключение между фигурами на рабочем поле.

Вариант 20. Игра "Тетрис"

Разработать упрощенный вариант этой популярной компьютерной игры. Игровое поле в форме стакана состоит из прямоугольных клеток и имеет границы слева, справа, снизу. Через равные промежутки времени в стакан сверху падают фигуры. Каждая фигура состоит из четырех соприкасающихся прямоугольных элементов. Возможные формы фигур: "кубик", линия, буква "Г", зеркальная буква "Г", буква "S", зеркальная буква "S", буква "Т". Во время падения фигура может быть подвинута пользователем влево и вправо, а также повернута на 90 градусов одним нажатием клавиши. Фигура останавливает движение, если ее продвижению вниз мешают ранее упавшие фигуры. Если какая-либо строка ячеек стакана оказывается полностью заполненной элементами фигур, то она удаляется, а все элементы фигур, расположенные выше, опускаются на одну строку вниз. Игра заканчивается если очередная падающая фигура не может двигаться. Различные формы падающих фигур генерируются с равной вероятностью.

Вариант 21. Моделирование движения молекул газа

Выполнить моделирование поведения молекул в прямоугольном сосуде, разделенном подвижной вертикальной перегородкой с отверстием. Молекулы отражаются от стенок и друг от друга без потери энергии. Пользователь может изменять величину отверстия в перегородке и перемещать перегородку в горизонтальном направлении.

Вариант 22. Игра "Ханойская башня"

Имеются три кольца и N дисков разного размера, перенумерованных от 1 до N в порядке возрастания их размеров. Сначала все диски надеты на первый колечко так, что их размеры уменьшаются снизу вверх. Цель игры состоит в том, чтобы перенести все диски на третий колечко, соблюдая при этом следующие условия: диски можно переносить только по одному, меньший диск нельзя ставить на больший диск. Разрабатываемое программное средство должно обеспечить выбор пользователем перемещаемого диска и места его перемещения, наглядное отображение процесса перемещения дисков, возможность выполнения обратных ходов на любое количество шагов назад.

Вариант 23. Моделирование образования тени

Цель данного задания - создание средства исследования формирования тени от нескольких объектов при наличии нескольких источников света. Экспериментальная установка представляет собой прямоугольное поле. Вдоль одной стороны поля располагаются точечные источники света. Противоположная сторона поля является экраном. На поле располагаются непрозрачные заслонки различных размеров, тень от которых формируется на экране. Пользователь может при помощи курсора выбирать либо источники света, либо заслонки. Выбранный источник может быть перемещен вдоль линии источников. Выбранная заслонка может быть перемещена в любом направлении, а также может быть изменен ее размер. При каждом новом расположении источников и заслонок изменяется диаграмма освещенности экрана.

Вариант 24. Игра "Зенитная установка"

На поверхности расположена подвижная зенитная установка, которая может стрелять вертикально вверх снарядами. На некотором уровне высоты с двух сторон в произвольные

моменты времени появляются самолеты, пролетающие над установкой. Из самолетов случайным образом выпадают бомбы, летящие вниз (учесть при этом скорость самолета, из которого они выпали). Пользователь управляет горизонтальным перемещением зенитной установки и стрельбой. Самолеты и бомбы уничтожаются при попадании в них снарядов. Игра заканчивается по команде пользователя, либо при попадании бомбы в установку.

Вариант 25. Модель регулятора уровня жидкости

Выполнить наглядное моделирование процесса регулирования уровня жидкости в емкости. В емкость через трубу сверху поступает жидкость, и через трубу снизу жидкость отводится. В отводящей трубе установлена заслонка. В емкости установлены два датчика уровня. Первый датчик подает сигнал "открыть заслонку", если он погружен в жидкость. Второй датчик вырабатывает сигнал "закрыть заслонку", если он не погружен в жидкость. Пользователю предлагается возможность изменять высоту расположения датчиков и производительность потока из подводящей трубы. Во время эксперимента в цифровой форме отображаются: производительность потока жидкости, количество жидкости в емкости.

Вариант 26. Моделирование пассажиропотока

По кольцевому маршруту движутся два автобуса в противоположном направлении. На маршруте имеются остановки. На каждую остановку с заданной интенсивностью прибывают пассажиры. Вместимость автобуса ограничена, поэтому по прибытию автобуса не все пассажиры могут в него попасть. Каждый пассажир с равной вероятностью выходит на любой последующей остановке и больше в автобус не заходит. Параметрами модели являются количество остановок и интенсивность притока пассажиров на каждую из них. В процессе моделирования следует отражать числами общее количество пассажиров на каждой остановке и в каждом автобусе, максимальное время ожидания на остановке для текущего момента времени. Сопровождать моделирование схематическим изображением процесса.

Вариант 27. Моделирование броуновского движения.

Моделировать движение молекул и нескольких крупных частиц. Количество молекул и количество частиц задаются перед каждым вариантом моделирования. Для каждой крупной частицы указывается ее масса.

Вариант 28. Моделирование криминальной системы.

В моделируемом пространстве хаотично двигаются «труженики», «воры» и «полицейские». В начальный момент времени каждый «труженик» имеет один кошелек. При встрече участников «вор» отбирает все кошельки у «труженика», «полицейский» отбирает все кошельки у «вора», «полицейский» отдает все кошельки «труженику». При моделировании отображать для каждого участника количество имеющихся у него кошельков, отдельно показывать общее количество кошельков у «тружеников», «воров» и «полицейских».

Фонд оценочных материалов (ФОМ) для проведения аттестации уровня сформированности компетенций обучающихся по дисциплине оформляется отдельным документом.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Книгообеспеченность

Наименование литературы: автор, название, вид издания, издательство	Год издания	КНИГООБЕСПЕЧЕННОСТЬ
		Наличие в электронном каталоге ЭБС
Основная литература		
1. Страуструп Б., Язык программирования С++ для профессионалов / Страуструп Б. - М. : Национальный Открытый Университет "ИНТУИТ", 2016.	2016	<URL : https://www.studentlibrary.ru/book/intuit_418.html >
2. Сvirкин М.В., Программирование под Windows в среде Visual C++ 2005 / Сvirкин М.В., Чуркин А.С. -	2016	<URL : https://www.studentlibrary.ru/book/intuit_28 >

М.: Национальный Открытый Университет "ИНТУИТ", 2016.		5.html>
3. Дубов, И.Р. Языки программирования. Объектно-ориентированное программирование: Учеб. пособие / И.Р. Дубов, В.И. Быков; Владим. гос. ун-т. им. А. Г. и Н. Г. Столетовых. – Владимир: Изд-во ВлГУ, 2017. – 100 с.	2018	<URL:http://dspace.www1.vlsu.ru/bitstream/123456789/7245/1/01731.pdf>
4. Барышникова М.Ю., Основы программирования на С/С++. Ч. 2 : Учеб. пособие / Барышникова М.Ю., Силантьева А.В. - М. : Издательство МГТУ им. Н. Э. Баумана, 2007. - 70 с. - ISBN 978-5-7038-2939-4	2007	< URL : https://www.studentlibrary.ru/book/ISBN9785703829394.html>
Дополнительная литература		
1. Снетков В.М., Практикум прикладного программирования на MFC и C++ в среде VS.NET / Снетков В.М. - М.: Национальный Открытый Университет "ИНТУИТ", 2016.	2016	<URL : https://www.studentlibrary.ru/book/intuit_259.html>
2. Эффективное использование C++. 55 верных способов улучшить структуру и код ваших программ / Мэйерс С. - М. : ДМК Пресс, 2008.	2008	< URL : https://www.studentlibrary.ru/book/ISBN5940743048.html>
3. Широков А.И., Стандартизация, сертификация и оценка качества программного обеспечения : учеб. пособие / А.И. Широков, Е.П. Потоцкий. - М. : МИСиС, 2013. - 208 с. - ISBN 978-5-87623-722-4	2013	< URL : https://www.studentlibrary.ru/book/ISBN9785876237224.html>

6.2. Периодические издания

Журналы (<https://elibrary.ru/>):

1. Вестник компьютерных и информационных технологий
2. Вычислительные технологии
3. Известия вузов: электроника
4. Радиотехнические и телекоммуникационные системы

6.3. Интернет-ресурсы

Центр дистанционного образования <https://cs.cdo.vlsu.ru/>
ЭБС Консультант студента <https://www.studentlibrary.ru/>

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для реализации данной дисциплины имеются специальные помещения для проведения занятий лекционного типа, занятий практического/лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы. Лабораторные работы проводятся в компьютерном классе ауд. 401.

Рабочую программу составил Шарафеддин М.А. доц. Каф. ВТиСУ

(ФИО, должность, подпись)

Рецензент

(представитель работодателя)

Генеральный директор ООО "Диаграмма" Протягов И.В.

(место работы, должность, ФИО, подпись)

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ВТиСУ

Протокол № 1 от 31.08.2021 года

Заведующий кафедрой

(ФИО, подпись)

Ланцов В.Н.

Рабочая программа рассмотрена и одобрена

на заседании учебно-методической комиссии направления 09.03.01

Протокол № 1 от 31.08.2021 года

Председатель комиссии

Ланцов В.Н. зав. каф. ВТиСУ

(ФИО, должность, подпись)

**ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ**

Рабочая программа одобрена на 20 22 / 20 23 учебный года

Протокол заседания кафедры № 1 от 29.08.22 года

Заведующий кафедрой  Куликов К.В.

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой _____

Рабочая программа одобрена на 20 ____ / 20 ____ учебный года

Протокол заседания кафедры № ____ от ____ года

Заведующий кафедрой _____