

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Смоленский государственный университет»  
Кафедра прикладной математики и информатики

«Утверждаю»  
Проректор по учебно-  
методической работе  
\_\_\_\_\_ Ю.А. Устименко  
«23» июня 2022 г.

**Рабочая программа дисциплины  
Б1.В.01.02 Программирование**

Направление подготовки: **44.03.05 Педагогическое образование (с двумя профилями подготовки)**

Направленность (профиль): **Математика, Информатика**

Форма обучения: очная

Курс – 3, 4

Семестр – 5, 6, 7

Всего зачетных единиц – 11, часов – 396

Форма отчетности: зачет – 5 семестр, экзамен – 6, 7 семестр

Программу разработал  
кандидат педагогических наук, доцент Козлов С.В.

Одобрена на заседании кафедры  
«16» июня 2022 г., протокол № 10

Заведующий кафедрой \_\_\_\_\_ С.В. Козлов

Смоленск  
2022

## 1. Место дисциплины в структуре ОП

Дисциплина «Программирование» является дисциплиной вариативной части подготовки бакалавра направления 44.03.05 Педагогическое образование (с двумя профилями подготовки). Она изучается в 5, 6 и 7 семестрах и является одной из основных дисциплин цикла. Дисциплина продолжает блок подготовки по информатике бакалавра по направлению «Педагогическое образование» (профиль «Математика, информатика»). Она является предшествующей для дисциплины «Практикум по решению задач на ЭВМ».

При изучении данной дисциплины необходимы компетенции студентов, сформированные при изучении таких дисциплин, как «Основы информатики и вычислительной техники» и «Алгоритмы и структуры данных». Курс построен так, чтобы сформировать у студентов целостное представление о фундаментальных основах программирования и развить у них практические навыки программирования на компьютере.

Изучение курса основано на традиционных методах высшей школы, тесной взаимосвязи со смежными курсами, а также на использовании современных систем программирования.

## 2. Планируемые результаты обучения по дисциплине

Компетенция	Индикаторы достижения
<b>ПК-6.</b> Способен использовать научные знания в предметной области (информатика) в процессе формирования предметной компетенции обучающихся в рамках реализации основной общеобразовательной программы	<b>Знать:</b> назначение, структуру и содержание курса информатики, современное состояние и перспективы развития информатики как учебной дисциплины, ее место и роль, фундаментальное ядро современного школьного курса информатики, принципы построения методической системы обучения информатике, ее основных компонентов. <b>Уметь:</b> анализировать цели и содержание школьного курса информатики, проектировать образовательный процесс, использовать дидактический потенциал средств информационных технологий в реализации образовательного процесса по курсу информатики; <b>Владеть:</b> основными видами профессиональной деятельности учителя информатики, профессиональными навыками реализации методики обучения основным разделам курса информатики, современными информационно-коммуникационными средствами для эффективного осуществления профессиональной деятельности.
<b>ПК-8.</b> Способен использовать современные системные программные средства, разрабатывать моделирующие алгоритмы и реализовывать их на базе языков и пакетов прикладных программ моделирования	<b>Знать:</b> основные принципы и методики создания алгоритмов и программ для решения прикладных задач, основные среды для разработки программного обеспечения, базовые информационные технологии программные средства; <b>Уметь:</b> корректно использовать современные информационные технологии и программные средства, внедрять и адаптировать прикладное программное обеспечение; <b>Владеть:</b> навыками решения прикладных задач с применением современных программных средств, владеть современными

	языками программирования и методиками разработки и внедрения прикладного программного обеспечения.
--	--

### 3. Содержание дисциплины

#### 5 семестр

1. **Среда программирования C#.** Основные понятия визуального программирования: классы, поля, свойства, методы, объекты, события и сообщения. Окно интерфейса C#: меню, горячие клавиши, библиотека компонентов. Форма и окно редактирования. Вспомогательные средства редактирования (инспектор объектов, значок кода, обозреватель). Использование визуальных компонент. Простейшая программа.
2. **Разработка проекта. Обзор свойств.** Файловая структура проекта (головной файл, модуль, файл ресурсов). Размещение на форме визуальных компонент. Обзор наиболее общих свойств (имя, размеры и положение на экране, обрамление, цвет, активность и видимость, управление курсором, задание подсказок). Вывод данных: метки, панели (ширина, выравнивание). Ввод: однострочные и многострочные редакторы (данные, размер, изменяемость, переносы).
3. **Подпрограммы. Процедуры и функции.** Понятие подпрограммы. Включение текста подпрограммы в раздел описания. Локальные и глобальные переменные. Иерархия подпрограмм. Заголовок процедуры. Параметры – аргументы и параметры – функции. Оператор досрочного завершения. Примеры: вывод окончаний имён числительных, рациональная арифметика. Правила описания функции. Получение значения функции. Примеры вычисления функций: сумма двух чисел. Рекурсивные вызовы функций.
4. **Классификация событий.** Программные события (OnFormCreat). Понятие фокуса, передача фокуса. События, вызываемые мышью, порядок наступления событий. Обработка событий, параметры Sender, Button, Shift, X, Y. Последовательность событий клавиатуры, распознавание клавиш, параметры Key, Shift. Перехват клавиатуры KeyPreview.
5. **Управление формами.** Оформление внешнего вида формы, расположение на экране, строка-заголовок. Встроенные диалоговые окна, MessageBox.Show.
6. **Многострочные редакторы. Компоненты-диалоги.** Компонент TextBox. Программирование абзацев, маркеров, цвета и шрифта. Обмен данными с диском. Диалоги по обмену данными, графические окна, вывод на принтер.
7. **Разработка меню.** Технология разработки панелей меню. Создание текстового редактора.
8. **Методы визуального программирования. Абсолютный доступ к памяти.** Технология Drag & Drop. Принцип адресации памяти. Операторы доступа к памяти. Использование системных переменных (клавиатура, таймер). Организация видеопамати в текстовом режиме (операции с битами).
9. **Описание сложных типов данных.** Сложный тип как структура данных. Примеры: массив, строка. Раздел описаний. Описание типа пользователя. Описание записи. Работа с записями. Пример: комплексные числа. Использование строк, массивов и записей в качестве параметров. Совмещение данных. Обработка матриц.
10. **Создание и использование библиотек.** Правила написания библиотечного модуля. Раздел инициализации модуля. Компилирование текста модуля. Модуль рациональных чисел.
11. **Текстовые файлы. Куча.** Файлы последовательного доступа. Имя файла. Переменные текстового типа. Открытие файла. Чтение и запись данных. Закрытие файла. Анализ файловых ошибок. Переменные-указатели. Понятие кучи. Типизированные указатели. Размещение данных в куче. Удаление данных из кучи. Использование кучи для работы с большими массивами данных.
12. **Стек.** Понятие структуры данных, связь с объектами. Определение стека. Расширение операций над стеком. Формирование стека при помощи массива. Стек как объект. Задача анализа парности скобок.

13. **Очередь.** Определение очереди. Организация кольцевой очереди при помощи массива. Волновой алгоритм (движение по шахматной доске).
14. **Однонаправленный список.** Определение однонаправленного списка. Формирование списка на основе одномерного массива. Организация стека при помощи однонаправленного списка. Создание списка в динамической памяти. Сортировка в однонаправленном списке.
15. **Двунаправленный список.** Определение двунаправленного списка. Реализация списка в форме объекта. Организация очереди при помощи двунаправленного списка.
16. **Графы. Деревья. Пути и циклы.** Определение ориентированного и неориентированного графа. Взвешенный граф. Задание графа в матричной форме и в виде списка. Обходы графа (в глубину и ширину). Определение неориентированного дерева. Ориентация дерева. Формирование ориентированного дерева на основе одномерного массива. Проход от листьев к стволу. Задачи на использование дерева.

### 6 семестр

1. **Основы объектно-ориентированного программирования.** Современные технологии программирования. Математическая теория объектов. Понятие объекта. Конструирование программ на основе иерархии объектов. Программирование объектов.
2. **Принципы объектно-ориентированного программирования.** Инкапсуляция. Описание формального объекта. Построение экземпляра объекта. Наследование. Разработка объектов-потомков. Полиморфизм. Методы разработки объектов.
3. **Интерфейсные объекты.** Управляющие элементы, окна, диалоги. Механизмы передачи и обработки сообщений в объектно-ориентированных средах. Программирование модулей и интерфейсов в языке C#.
4. **Графические объекты.** Графический вывод на экран. Графические примитивы. Графика на битовой карте. Графика на форме. Цвет, режимы пера. Геометрические свойства цвета. Копирование изображений.
5. **Типизированные файлы.** Открытие файла. Чтение и запись данных. Закрытие файла. Анализ файловых ошибок. Особенности работы с типизированными файлами в C#. Преобразование текстового файла в типизированный файл. Дихотомический поиск в упорядоченном файле. Создание хэш-файла и поиск в нем данных.
6. **Обработка больших объемов данных.** Алгоритмы внешней сортировки. Алгоритмы слияния строго упорядоченных файлов. Алгоритмы слияния нестрого упорядоченных файлов.

### 7 семестр

1. **Реляционные базы данных.** Концептуальная модель предметной области. Логическая модель предметной области. Понятие базы данных. Определение взаимосвязи между элементами баз данных. Модели баз данных. Реляционная модель. Организация реляционной базы. Первичные и альтернативные ключи атрибутов данных. Отношения между таблицами. Типы связей. Ссылочная целостность. Приведение модели к требуемому уровню нормальной формы.
2. **Проектирование базы данных.** Архитектура СУБД. Обзор возможностей и особенностей различных СУБД. Методы хранения и доступа к данным. Работа с внешними данными с помощью технологии ODBC (ADO, BDE, JDBC). Объектно-ориентированное программирование в среде баз данных. Разработка макета базы данных.
3. **Создание БД, навигация по таблице.** Создание базы данных в среде языка объектно-ориентированного программирования C#. Связь с базой через компонент Table. Визуализация полей таблицы. Разработка и создание простейших СУБД.
4. **Фильтрация, индексирование и поиск данных.** Возможности фильтрации. Способы создания фильтров. Индексирование полей. Сортировка таблицы по столбцам.
5. **Введение в язык SQL.** Основы языка SQL. Связь с базой через компонент Query. Организация простейших SQL-запросов. Использование SQL для выборки данных из таблицы (операторы Select и From). Визуальный построитель запросов Builder.

6. **Запросы на SQL.** Основы языка SQL. Выполнение сложных запросов на SQL. Исключение повторяющихся значений (оператор Distinct). Использование оператора Like. Вычисляемые поля. Агрегатные функции. Организация вложенных запросов. Использование оператора Exists. Модификация данных (операторы Insert, Update, Delete). Создание и удаление таблиц баз данных. Создание и удаление индексов.
7. **Основы защиты информации.** Понятие информационной безопасности. Категории информационной безопасности (доступность, целостность, конфиденциальность). Наиболее распространённые угрозы. Угрозы доступности: примеры. Вредоносное ПО. Основные угрозы целостности и конфиденциальности. Уровни безопасности.
8. **Методы защиты информации. Шифрование данных.** Шифрование информации. Криптографические основы безопасности: основные понятия. Односторонняя функция. Хранение паролей в памяти компьютера. Криптография с открытым ключом. Хеш-функции. Шифры Цезаря, Вижинера, Диффи-Хеллмана, Эль-Гамала, RSA. Цифровая подпись.

#### 4. Тематический план

##### 5 семестр

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Среда программирования C#	8	2	–	4	2
2	Разработка проекта. Обзор свойств	10	2	–	4	4
3	Подпрограммы. Процедуры и функции	8	2	–	2	4
4	Классификация событий	8	2	–	2	4
5	Управление формами	8	2	–	2	4
6	Многострочные редакторы. Компоненты-диалоги	8	2	–	2	4
7	Разработка меню	10	2	–	4	4
8	Методы визуального программирования. Абсолютный доступ к памяти	10	2	–	4	4
9	Описание сложных типов данных	8	2	–	2	4
10	Создание и использование библиотек	8	2	–	2	4
11	Текстовые файлы. Куча	8	2	–	2	4
12	Стек	10	2	–	2	6
13	Очередь	10	2	–	2	6
14	Однонаправленный список	10	2	–	2	6
15	Двунаправленный список	10	2	–	2	6
16	Графы. Деревья. Пути и циклы	10	2	–	2	6
ИТОГО 5 семестр		144	32	–	40	72

### 6 семестр

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Основы объектно-ориентированного программирования	16	4	–	4	8
2	Принципы объектно-ориентированного программирования	18	4	–	6	8
3	Интерфейсные объекты	18	4	–	4	10
4	Графические объекты	18	4	–	4	10
5	Типизированные файлы	18	4	–	4	10
6	Обработка больших объемов данных	20	6	–	4	10
ИТОГО 6 семестр		108	26	–	26	29+27

### 7 семестр

№ п/п	Разделы и темы	Всего часов	Формы занятий			
			лекции	практические занятия	лабораторные занятия	самостоятельная работа
1	Реляционные базы данных	18	4	–	4	10
2	Проектирование базы данных	20	4	–	4	12
3	Создание БД, навигация по таблице	18	2	–	4	12
4	Фильтрация, индексирование и поиск данных	18	2	–	4	12
5	Введение в язык SQL	16	2	–	4	10
6	Запросы на SQL	20	4	–	4	12
7	Основы защиты информации	16	2	–	4	10
8	Методы защиты информации. Шифрование данных	18	4	–	4	10
ИТОГО 7 семестр		144	24	–	32	61+27
ИТОГО		396	82	–	98	162+54

## 5. Виды образовательной деятельности

### Занятия лекционного типа

### 5 семестр

1. **Среда программирования C#.** Основные понятия визуального программирования: классы, поля, свойства, методы, объекты, события и сообщения. Окно интерфейса C#: меню, горячие клавиши, библиотека компонентов. Форма и окно редактирования. Вспомогательные средства редактирования (инспектор объектов, значок кода, обозреватель). Использование визуальных компонент. Простейшая программа.

2. **Разработка проекта. Обзор свойств.** Файловая структура проекта (головной файл, модуль, файл ресурсов). Размещение на форме визуальных компонент. Обзор наиболее общих свойств (имя, размеры и положение на экране, обрамление, цвет, активность и видимость, управление курсором, задание подсказок). Вывод данных: метки, панели (ширина, выравнивание). Ввод: однострочные и многострочные редакторы (данные, размер, изменяемость, переносы).

3. **Подпрограммы. Процедуры и функции.** Понятие подпрограммы. Включение текста подпрограммы в раздел описания. Локальные и глобальные переменные. Иерархия подпрограмм. Заголовок процедуры. Параметры – аргументы и параметры – функции. Оператор досрочного завершения. Примеры: вывод окончаний имён числительных, рациональная арифметика. Правила описания функции. Получение значения функции. Примеры вычисления функций: сумма двух чисел. Рекурсивные вызовы функций.

4. **Классификация событий.** Программные события (OnFormCreat). Понятие фокуса, передача фокуса. События, вызываемые мышью, порядок наступления событий. Обработка событий, параметры Sender, Button, Shift, X,Y. Последовательность событий клавиатуры, распознавание клавишей, параметры Key, Shift. Перехват клавиатуры KeyPreview.

5. **Управление формами.** Оформление внешнего вида формы, расположение на экране, строка-заголовок. Встроенные диалоговые окна, MessageBox.Show.

6. **Многострочные редакторы. Компоненты-диалоги.** Компонент TextBox. Программирование абзацев, маркеров, цвета и шрифта. Обмен данными с диском. Диалоги по обмену данными, графические окна, вывод на принтер.

7. **Разработка меню.** Технология разработки панелей меню. Создание текстового редактора.

8. **Методы визуального программирования. Абсолютный доступ к памяти.** Технология Drag & Drop. Принцип адресации памяти. Операторы доступа к памяти. Использование системных переменных (клавиатура, таймер). Организация видеопамати в текстовом режиме (операции с битами).

9. **Описание сложных типов данных.** Сложный тип как структура данных. Примеры: массив, строка. Раздел описаний. Описание типа пользователя. Описание записи. Работа с записями. Пример: комплексные числа. Использование строк, массивов и записей в качестве параметров. Совмещение данных. Обработка матриц.

10. **Создание и использование библиотек.** Правила написания библиотечного модуля. Раздел инициализации модуля. Компилирование текста модуля. Модуль рациональных чисел.

11. **Текстовые файлы. Куча.** Файлы последовательного доступа. Имя файла. Переменные текстового типа. Открытие файла. Чтение и запись данных. Закрытие файла. Анализ файловых ошибок. Переменные-указатели. Понятие кучи. Типизированные указатели. Размещение данных в куче. Удаление данных из кучи. Использование кучи для работы с большими массивами данных.

12. **Стек.** Понятие структуры данных, связь с объектами. Определение стека. Расширение операций над стеком. Формирование стека при помощи массива. Стек как объект. Задача анализа парности скобок.

13. **Очередь.** Определение очереди. Организация кольцевой очереди при помощи массива. Волновой алгоритм (движение по шахматной доске).

14. **Однонаправленный список.** Определение однонаправленного списка. Формирование списка на основе одномерного массива. Организация стека при помощи однонаправленного списка. Создание списка в динамической памяти. Сортировка в однонаправленном списке.

15. **Двунаправленный список.** Определение двунаправленного списка. Реализация списка в форме объекта. Организация очереди при помощи двунаправленного списка.

16. **Графы. Деревья. Пути и циклы.** Определение ориентированного и неориентированного графа. Взвешенный граф. Задание графа в матричной форме и в виде списка. Обходы графа (в глубину и ширину). Определение неориентированного дерева. Ориентация дерева Формирование ориентированного дерева на основе одномерного массива. Проход от листьев к стволу. Задачи на использование дерева.

## 6 семестр

1. **Основы объектно-ориентированного программирования.** Современные технологии программирования. Математическая теория объектов. Понятие объекта. Конструирование программ на основе иерархии объектов. Программирование объектов.

2. **Принципы объектно-ориентированного программирования.** Инкапсуляция. Описание формального объекта. Построение экземпляра объекта. Наследование. Разработка объектов-потомков. Полиморфизм. Методы разработки объектов.

3. **Интерфейсные объекты.** Управляющие элементы, окна, диалоги. Механизмы передачи и обработки сообщений в объектно-ориентированных средах. Программирование модулей и интерфейсов в языке C#.

4. **Графические объекты.** Графический вывод на экран. Графические примитивы. Графика на битовой карте. Графика на форме. Цвет, режимы пера. Геометрические свойства цвета. Копирование изображений.

5. **Типизированные файлы.** Открытие файла. Чтение и запись данных. Закрытие файла. Анализ файловых ошибок. Особенности работы с типизированными файлами в C#. Преобразование текстового файла в типизированный файл. Дихотомический поиск в упорядоченном файле. Создание хэш-файла и поиск в нем данных.

6. **Обработка больших объемов данных.** Алгоритмы внешней сортировки. Алгоритмы слияния строго упорядоченных файлов. Алгоритмы слияния нестрого упорядоченных файлов.

## 7 семестр

1-2. **Реляционные базы данных.** Концептуальная модель предметной области. Логическая модель предметной области. Понятие базы данных. Определение взаимосвязи между элементами баз данных. Модели баз данных. Реляционная модель. Организация реляционной базы. Первичные и альтернативные ключи атрибутов данных. Отношения между таблицами. Типы связей. Ссылочная целостность. Приведение модели к требуемому уровню нормальной формы.

3-4. **Проектирование базы данных.** Архитектура СУБД. Обзор возможностей и особенностей различных СУБД. Методы хранения и доступа к данным. Работа с внешними данными с помощью технологии ODBC (ADO, BDE, JDBC). Объектно-ориентированное программирование в среде баз данных. Разработка макета базы данных.

5. **Создание БД, навигация по таблице.** Создание базы данных в среде языка объектно-ориентированного программирования C#. Связь с базой через компонент Table. Визуализация полей таблицы. Разработка и создание простейших СУБД.

6. **Фильтрация, индексирование и поиск данных.** Возможности фильтрации. Способы создания фильтров. Индексирование полей. Сортировка таблицы по столбцам.

7. **Введение в язык SQL.** Основы языка SQL. Связь с базой через компонент Query. Организация простейших SQL-запросов. Использование SQL для выборки данных из таблицы (операторы Select и From). Визуальный построитель запросов Builder.

8-9. **Запросы на SQL.** Основы языка SQL. Выполнение сложных запросов на SQL. Исключение повторяющихся значений (оператор Distinct). Использование оператора Like. Вычисляемые поля. Агрегатные функции. Организация вложенных запросов. Использование оператора Exists. Модификация данных (операторы Insert, Update, Delete). Создание и удаление таблиц баз данных. Создание и удаление индексов.

10. **Основы защиты информации.** Понятие информационной безопасности. Категории информационной безопасности (доступность, целостность, конфиденциальность). Наиболее распространённые угрозы. Угрозы доступности: примеры. Вредоносное ПО. Основные угрозы целостности и конфиденциальности. Уровни безопасности.

11-12. **Методы защиты информации. Шифрование данных.** Шифрование информации. Криптографические основы безопасности: основные понятия. Односторонняя функция. Хранение паролей в памяти компьютера. Криптография с открытым ключом. Хеш-функции. Шифры Цезаря, Вижинера, Диффи-Хеллмана, Эль-Гамала, RSA. Цифровая подпись.



## Занятия семинарского типа

Не предусмотрены.

## Практические занятия

Не предусмотрены.

## Лабораторные занятия

### 5 семестр

#### Лабораторное занятие №1.

*Среда программирования C#*

Продемонстрировать ввод и вывод данных в C#.

```
// Вычислить сумму двух чисел
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Привет");
            double x, y, z;
            Console.WriteLine("Введите первое число");
            x = Convert.ToInt16(Console.ReadLine());
            Console.WriteLine("Введите второе число");
            y = Convert.ToInt16(Console.ReadLine());
            z = x + y;
            Console.WriteLine("Сумма чисел равна={0}",s);
            Console.WriteLine("Сумма чисел равна="+s);
            Console.ReadLine();
        }
    }
}
```

Задания.

1. Вычислить площадь прямоугольника по двум его сторонам.
2. Вычислить периметр и площадь треугольника по заданным его сторонам.
3. Вычислить площадь круга, длину окружности и объем шара заданного радиуса R.
4. Вычислить путь, пройденный катером по и против течения, если известны его скорость, скорость реки и время движения по течению и против течения.
5. Две точки заданы своими координатами в пространстве. Вычислить расстояние между ними.

В чем состоит отличие команд Convert и Parse?

Решение минимум двух задач оформить с помощью метода Parse.

#### Лабораторное занятие №2.

*Среда программирования C#*

Задания.

1. Вычислить объем и площадь полной поверхности прямоугольного параллелепипеда со сторонами  $a$ ,  $b$ ,  $c$ .
2. Заданы координаты трех вершин треугольника. Найти его периметр и площадь.
3. Дана сторона равностороннего треугольника. Найти его площадь, периметр, высоты, радиус вписанной и описанной окружностей.
4. Дано целое положительное число  $n$ . Определить последнюю цифру этого числа.
5. Найти сумму цифр заданного трехзначного числа.

### **Лабораторное занятие №3.**

#### *Разработка проекта. Обзор свойств*

Задания.

1. Дано четырехзначное число  $a$ . Составить целое число  $b$ , полученное выписыванием в обратном порядке цифр заданного числа.
2. Найти количество натуральных чисел, не превосходящих заданного числа  $n$ , которые делятся на 7, но не делятся на 5.
3. Известны стороны  $a$ ,  $b$  и  $c$  треугольника. Вычислить его высоты.
4. Задать положительное вещественное число  $x$ . Составить число  $y$  из цифр сотен и десятков числа  $x$ .
5. Вычислить значение произвольного выражения от 4 переменных, которое содержит арифметические операции, возведение в степень, извлечение корня  $n$ -ой степени, логарифм, модуль и тригонометрические функции.

### **Лабораторное занятие №4.**

#### *Разработка проекта. Обзор свойств*

Задания.

- Задача № 1. Найти количество натуральных чисел, не превосходящих заданного числа  $N$ , которые
- 1) делятся на 7, но не делятся на 5
  - 2) делятся и на 7, и на 5
  - 3) не делятся ни на 7, ни на 5
  - 4) делятся хотя бы на 7 или на 5

Задача № 2. Задать положительное вещественное число  $X$ . Составить целое число  $Y$  из цифр

- 1) сотен и десятых  $X$
- 2) десятков и сотых  $X$
- 3) десятков и десятых  $X$
- 4) единиц и сотых  $X$

Задача № 3. Задать положительное целое число  $X$ . Указать целое  $Y$ , которое является

- 1) наибольшим полным квадратом, не превосходящим  $X$
- 2) наибольшей целой степенью 10, не превосходящей  $X$
- 3) наименьшим кратным 10, которое больше или равно  $X$
- 4) наименьшим, квадрат которого больше или равен  $X$

Задача № 4. Написать программу, вычисляющую значение периодической функции  $y(x)$  в произвольной точке  $x$ , если на промежутке  $[a;b]$ , который составляет один её период, значения функции вычисляются по формуле

- 1)  $[-2; 2]$ ,  $y = |x|+1$
- 2)  $[-5;-1]$ ,  $y = |1/x + 0.6|$
- 3)  $[-1; 2]$ ,  $y = \sin|\text{Pi}*x|$
- 4)  $[-2; 1]$ ,  $y = \cos|\text{Pi}*x|$

## Лабораторное занятие №5.

### Подпрограммы. Процедуры и функции

Задания.

Задача № 1. Даны координаты концов двух отрезков  $[x_1, x_2]$ ,  $[x_3, x_4]$ . Верно ли, что

- 1) отрезки пересекаются
- 2) отрезки не пересекаются
- 3) первый отрезок вложен во второй
- 4) отрезки имеют единственную общую точку

Задача № 2. Проверить, что натуральное число  $N$

- 1) является 3-значным палиндромом
- 2) образовано четырьмя одинаковыми цифрами
- 3) является счастливым четырёхзначным
- 4) образовано тремя цифрами, составляющими арифметическую прогрессию

Задача № 3. Сообщить является ли вещественное значение  $X$

- 1) целой степенью числа 2
- 2) корнем целой степени из 2
- 3) синусом угла кратного 45 градусам
- 4) тангенсом угла кратного 30 градусам

Задача № 4. Написать программу, вычисляющую значение функции  $y(x)$  или сообщаящей об ошибке вычислений

- 1)  $y(x) = \text{ctg}(x/(x-1))$
- 2)  $y(x) = \ln(\sin(1/x))$
- 3)  $y(x) = \text{sqrt}(\text{tg}(x))$
- 4)  $y(x) = \ln(\text{sqrt}(1-x))$

## Лабораторное занятие №6.

### Классификация событий

Задания.

Задача № 1. Даны четыре логических значения  $b_1, b_2, b_3, b_4$ . Записать условие того, что:

- 1) более двух значений являются истинными
- 2) хотя бы два значения истинны
- 3) точно два значения истинны
- 4) три значения являются одинаковыми

Задача № 2. Ввести координаты точки  $(x, y)$  и проверить, что точка принадлежит фигуре:

- 1) Ширина фигуры равна 3 единицы, высота – 5 единиц
- 2) Ширина фигуры равна 2 единицы, высота – 6 единиц
- 3) Ширина фигуры равна 4 единицы, высота – 5 единиц
- 4) Ширина фигуры равна 3 единицы, высота – 6 единиц

Начало координат совпадает с верхним левым углом фигуры

Задача № 3. При заданном значении  $X$  выразить значение  $Y$  из уравнения

- 1)  $Y^3 X^2 + 3 Y^3 X = 4$
- 2)  $Y^5 X - 2 Y^5 X^2 = 3$
- 3)  $Y^3 X^3 + 2 Y^3 X = 3$

$$4) Y^5 X^3 - 3 Y^5 X^3 = 4$$

Задача № 4. Среди трёхзначных полных квадратов найти

- 1) последний, у которого совпадают крайние цифры
- 2) первый, у которого совпадают две первые цифры
- 3) последний, у которого совпадают две последние цифры
- 4) первый, у которого какая то крайняя цифра равна 5

### Лабораторное занятие №7.

#### Управление формами

Задания.

Задача № 1. Разложите натуральное число N на простые множители и определите

- 1) сколько раз встречается множитель 3 в произведении
- 2) сколько различных множителей в произведении
- 3) наименьший множитель произведения
- 4) наибольший множитель произведения

Задача № 2. Каждая буква замещает некоторую десятичную цифру 0 - 9.

Расшифруйте равенство и найдите наибольшую СКОРОСТЬ, если

- 1) СКО\*Р=ОСТЬ [998\*9=8982]
- 2) С\*КОР=ОСТЬ [9\*989=8901]
- 3) СКОР:ОС=ТЬ [7191:9=799]
- 4) СКОР:ОСТ=Ь [8892:988=9]

Задача № 3. Найдите положительный корень уравнения с точностью d

- 1)  $x = \cos(x)$
- 2)  $\ln(x) = -x$
- 3)  $\sin(x) = x - 1$
- 4)  $\text{Exp}(x) = x + 2$

Приближённое значение корня должно отличаться от точного значения более, чем на d.

Задача № 4.

1) Складывая салфетки, официант следит, чтобы над белой салфеткой лежала красная или синяя, над синей - только белая, а над красной - какая угодно. Сколькими способами официант может сложить N салфеток?

2) Дизайнер связывает гирлянду фонариков. Причём, вслед за круглым фонариком он может разместить квадратный или фигурный, за квадратным - только круглый, а за фигурным - любой. Сколькими способами дизайнер может связать N фонариков?

3) Библиотекарь расставляет книги на полку. Правее толстой книги он может поставить тонкую или среднюю, правее средней - толстую или тонкую, а правее тонкой - только среднюю. Сколькими способами библиотекарь может расставить N книг?

4) Кукольник вытаскивает матрёшек в виде колобков, поросят и зайцев. В колобок он может вставить поросёнка или зайца, в поросёнка - только зайца, а в зайца - колобка или поросёнка. Сколькими способами кукольник может разместить N матрёшек?

### Лабораторное занятие №8.

#### Многострочные редакторы. Компоненты-диалоги

Задания.

Задача № 1. Составить программу для поиска целочисленных неотрицательных решений (x,y) уравнения с параметром a

- 1)  $X^2 + 5*Y^2 = a$
- 2)  $3X^2 + Y^2 = a$
- 3)  $X^2 + Y^2 + Y = a$

$$4) \quad X^2 + X^2 + 2Y = a$$

Задача № 2. Проверить является ли логическое выражение тождеством.

$$1) \quad \overline{(X \text{ and } Y)} \text{ and } (X \text{ or } Y) = X \text{ xor } Y$$

$$2) \quad \overline{(X \text{ or } Y)} = \bar{X} \text{ and } \bar{Y}$$

$$3) \quad (\bar{X} \text{ and } \bar{Y}) \text{ or } (X \text{ and } Y) = X \text{ xor } Y$$

$$4) \quad (X \text{ xor } Y) \text{ and } \bar{X} = X \text{ and } Y$$

Задача № 3. Известно, что хотя бы один из подозреваемых А,В,С виновен в преступлении.

Дополнительно установлено, что

- 1) Если А виновен, а В невиновен, то С виновен;  
С всегда находится в компании с кем-то из А или В;  
А не "работает" вместе с С
- 2) Если А виновен, то у него был хотя бы один сообщник;  
Если В виновен, то виновен и С;  
Если С невиновен, то невиновен кто-то ещё;
- 3) Если виновен А, то у него был один сообщник;  
Если виновен В, то у него было два сообщника;
- 4) А не работает один;  
Если В виновен, то виновны А или С;  
Если А виновен, то В невиновен;

Чья виновность или невиновность не вызывает сомнений?

Задача № 4.

1) Два автомата последовательно анализируют двузначные числа 10..99, написанные на упаковках изделий. Один из них увеличивает счетчик, если сумма цифр числа оказывается больше 12, другой - увеличивает тот же счетчик, если замечает среди цифр восьмёрку, но при условии, что первый автомат не изменил показания счётчика. Какое число будет написано на упаковке, когда значение счетчика станет равным N?

2) Дети играют в "хлопки". Они по очереди считают от 1 и хлопают в ладоши, если число делится на 3 или кончается на 3. При каком счёте прозвучит хлопок N?

3) По конвейеру движутся пронумерованные от 10 до 99 пакеты с изделиями. Для проверки качества извлекается каждый 9-й и те, что содержат цифру 5. Какой двузначный номер будет иметь пакет, извлечённый N-м по порядку?

4) Каждого третьего по списку ученика направляют на уборку территории, а каждого девятого из списка делают звеньевым соответствующей уборочной бригады. Сколько учеников списка не будут участвовать в уборке, когда назначат N-го звеньевого?

### Лабораторное занятие №9-10.

#### Разработка меню

Задания.

Задача № 1. Задан 32-элементный массив больших русских букв от А до Я (без Ё).

Требуется выбрать поддиапазон [X, Y] массива и, не меняя порядка оставшихся элементов, сдвинуть его в

- 1) начало массива, сохранив порядок элементов
- 2) конец массива, сохранив порядок элементов
- 3) начало массива, переставляя элементы в обратном порядке
- 4) конец массива, переставляя элементы в обратном порядке

Результат изменений показать на экране

Задача № 2. Сформировать случайным образом N-элементный массив, заполненный однозначными целыми числами разных знаков. Задать X и выяснить

- 1) сколько раз повторяется X в массиве
- 2) под каким номером стоит первый X
- 3) под каким номером стоит последний X
- 4) сколько имеется элементов не равных X

Задача № 3. Сформировать случайным образом целочисленный массив из 10 натуральных двузначных чисел. Найти среднее значение S и

- 1) число элементов, меньших S
- 2) элемент, наиболее близкий к S
- 3) элемент, наиболее удалённый от S
- 4) число элементов, больших S

Задача № 4. Имеется числовой массив A. Скопировать его элементы в массив B, отбрасывая те, которые нарушают порядок

- 1) возрастания
- 2) убывания
- 3) знакочередования
- 4) стремление к 0 (по модулю)

Показать оба массива

### **Лабораторное занятие №11-12.**

*Методы визуального программирования. Абсолютный доступ к памяти*

Задания.

Задача № 1. Заданный текст преобразуйте в строку, где удалены все

- 1) ".", а пробелы заменены на "\_!\_"
- 2) " ", а "-" заменены на "\*\*\*"
- 3) "-", а малые "z" исправлены на "(Z)"
- 4) "\*\*", а большие "Z" исправлены на "-z-"

Результат изменений показать на экране

Задача № 2. Ввести строку, задающее число в системе счисления по основанию P. Найти десятичное значение числа. Если строка содержит посторонний символ, то необходимо указать место ошибки

- 1) P=2; 2) P=3; 3) P=4; 4) P=5.

Задача № 3. Ввести латинский текст и выписать различные слова, которые

- 1) начинаются большой буквой
- 2) начинаются цифрой
- 3) заканчиваются малой буквой
- 4) взяты в кавычки, например, "cat"

Задача № 4.

Найти значение выражения, которое хранится в строке в виде

- 1) суммы двух целых чисел
- 2) разности двух целых чисел
- 3) произведения двух целых чисел
- 4) частного двух целых чисел

### **Лабораторное занятие №13.**

#### *Описание сложных типов данных*

Задания.

1. Дан текст. Вывести все слова, которые начинаются с заданной буквы.
2. Определить длину самого короткого слова в заданном тексте.
3. Заменить все вхождения слова "Оля" на "Коля".
4. Найти сумму цифр в заданной строке.
5. Вывести на экран все трехбуквенные слова в заданном тексте.

### **Лабораторное занятие №14.**

#### *Создание и использование библиотек*

Задания.

1. Удалить среднюю букву при нечетной длине строки и две средних буквы при четной длине строки.
2. Даны две строки. Вывести буквы, встречающиеся и в той и в другой строках.
3. Дана строка символов. Выделить подстроку между первой и последней точкой.
4. Определить количество слов в строке, начинающихся и оканчивающихся одной и той же буквой.
5. Дан текст. Определить количество слов, являющихся палиндромами.

### **Лабораторное занятие №15.**

#### *Текстовые файлы. Куча*

Задания.

1. Разработать метод, вычисляющий среднее арифметическое двух заданных чисел.
2. Разработать метод для нахождения максимального значения четырех заданных чисел с помощью метода нахождения максимального значения двух заданных чисел.
3. Разработать метод вычисления факториала заданного целого числа N.
4. Разработать метод удаления из заданного текста заданной подстроки.
5. Разработать метод перевода заданного натурального десятичного числа N в двоичную систему счисления.

### **Лабораторное занятие №16.**

#### *Стек*

Задания.

1. Разработать методы вычисления НОД и НОК двух заданных натуральных чисел. Продемонстрировать различное написание методов.
2. Разработать методы сортировки заданного одномерного массива - "пузырьком", обменом, вставками и сортировку, определенную преподавателем.
3. Разработать методы работы с двумерными массивами: поиск наибольшего и наименьшего значений, количество положительных элементов, сумма элементов на главной диагонали, поворот матрицы на 90 градусов по часовой стрелке.

### **Лабораторное занятие №17.**

#### *Очередь*

Задания.

1. Разработать приложение для вычисления пути катера, если известны его скорость, скорость течения реки и время движения по течению и против течения.
2. Разработать приложение, которое по заданному значению одной величины вычисляет другие элементы равностороннего треугольника. В качестве элементов треугольника рассмотреть сторону, высоту, площадь, радиусы вписанной и описанной окружностей.
3. Разработать приложение, которое выполняет перевод массы из одних единиц измерения в другие. В качестве единиц измерения взять граммы, миллиграммы, килограммы, центнеры и тонны.

4. Разработать приложение, которое по заданным трем числам определяет, существует ли треугольник, а если существует то какого он вида: произвольный, равнобедренный, равносторонний, остроугольный, прямоугольный, тупоугольный.

5. Разработать приложение, которое для заданного четырехзначного числа вычисляет сумму и произведение его цифр, определяет, является ли число счастливым, подсчитывает сколько в числе четных цифр.

### **Лабораторное занятие №18.**

#### *Однонаправленный список*

Задания.

1. Разработать приложение, демонстрирующее работу с элементами управления CheckBox и RadioButton. Осуществить выбор элемента из списка CheckBox, выбрать один из параметров RadioButton и вывести сообщение в диалоговое окно с помощью метода MessageBox.Show.

2. Разработать приложение, которое демонстрирует работу простейшего калькулятора: выполнение арифметических действий, вычисление квадрата, квадратного корня и степени числа, вычисление модуля и значений тригонометрических функций.

3. Разработать приложение, демонстрирующее работу с целыми числами: вычисление суммы и произведения цифр числа, нахождение минимальной и максимальной цифр в числе, перестановка первой и последней цифр числа, является ли последовательность цифр числа возрастающей.

4. Разработать приложение, демонстрирующее создание целочисленного одномерного массива, указанной длины. Осуществить его заполнение с помощью датчика случайных чисел. Выбор диапазона осуществить при помощи элемента управления RadioButton, сгруппировав элементы множества на панели GroupBox. Вычислить среднее арифметическое или среднее гармоническое заданных чисел. Выбор осуществить с помощью элемента CheckBox, результат вывести в поле TextBox.

### **Лабораторное занятие №19.**

#### *Двунаправленный список*

Задания.

1. Разработать приложение, демонстрирующее работу с одномерным массивом: заполнение массива на форме; определение каких элементов в массиве больше положительных или отрицательных; формирование из исходного нового массива, в котором каждый элемент, стоящий на четном месте удвоен, а на нечетном утроен; нахождение в массиве наибольшего и наименьшего значений.

2. Разработать приложение, демонстрирующее работу с одномерным массивом: заполнение массива на форме; определение второго по величине наибольшего элемента с указанием его позиции в массиве; нахождение в массиве первого положительного и последнего отрицательного элементов; поменять местами наибольший четный и наименьший нечетный элементы.

3. Разработать приложение, демонстрирующее работу с одномерным массивом: отсортировать массив "пузырьком", обменом, вставками и своей сортировкой; отсортировать элементы, стоящие на четных местах по возрастанию, на нечетных по убыванию (с и без использования вспомогательных массивов).

4. Разработать приложение, демонстрирующее работу с двумерным массивом: заполнение массива на форме, вычисление суммы элементов массива, вычисление количества положительных элементов в массиве, определение каких элементов в массиве больше четных или кратных трем, вычисление суммы элементов в каждой строке, выяснение есть ли в каждом столбце массива нулевой элемент, вычисление произведения элементов массива кратных пяти.

5. Разработать приложение, демонстрирующее работу с двумерным массивом: нахождение суммы элементов на главной и побочной диагоналях, выше и ниже главной диагонали; поворот матрицы на 90, 180, 270 градусов по часовой стрелке, отображение матрицы относительно



горизонтальной, вертикальной оси симметрии, главной и побочной диагоналей (с выводом каждого нового вида матрицы).

6. Разработать приложение, демонстрирующее работу со строками: поиск вхождения слова в заданной строке, вывод слов строки в алфавитном порядке с указанием числа их встречаемости, упорядочивание слов строки в порядке, обратном алфавитному порядку.

### **Лабораторное занятие №20.** *Графы. Деревья. Пути и циклы*

Задания.

1. Сформировать стек из N натуральных чисел, просмотреть его и уменьшить нечетные его элементы на 1.

2. Сформировать очередь из N натуральных чисел, просмотреть ее и найти сумму второго и четвертого элементов очереди. Результат поместить с очередь.

В данных задачах разработать собственные методы инициализации, помещения и извлечения элемента и просмотра стека и очереди.

3. На клеточной карте заданы острова (-1) и водная поверхность. Пользователь вводит координаты исходной и конечной точки маршрута (движение возможно только по воде). Требуется проложить путь между заданными точками. Движение можно осуществлять по горизонтали либо по вертикали. Дополнительно определить длину маршрута и выделить путь цветом.

## **6 семестр**

### **Лабораторное занятие 1-5.**

#### *Основы объектно-ориентированного программирования* *Принципы объектно-ориентированного программирования*

Создать приложение, удовлетворяющее следующим требованиям:

1. Использовать возможности ООП: классы, наследование, полиморфизм, инкапсуляция.
2. Каждый класс должен иметь исчерпывающее смысл название и информативный состав.
3. Наследование должно применяться только тогда, когда это имеет смысл.
4. Классы должны быть грамотно разложены по пакетам.
5. Для хранения параметров инициализации использовать файлы.
6. Объединить модули программного приложения в единый интерфейс.

Стандартные темы для разработки многофункционального интерфейса на основе методологии объектно-ориентированного программирования:

1. **Цветочница.** Определить иерархию цветов. Создать несколько объектов-цветов. Собрать букет (используя аксессуары) с определением его стоимости. Провести сортировку цветов в букете на основе уровня свежести. Найти цветок в букете, соответствующий заданному диапазону длин стеблей.
2. **Новогодний подарок.** Определить иерархию конфет и прочих сладостей. Создать несколько объектов-конфет. Собрать детский подарок с определением его веса. Провести сортировку конфет в подарке на основе одного из параметров. Найти конфету в подарке, соответствующую заданному диапазону содержания сахара.
3. **Домашние электроприборы.** Определить иерархию электроприборов. Включить некоторые в розетку. Посчитать потребляемую мощность. Провести сортировку приборов в квартире на основе мощности. Найти прибор в квартире, соответствующий заданному диапазону параметров.
4. **Шеф-повар.** Определить иерархию овощей. Сделать салат. Посчитать калорийность. Провести сортировку овощей для салата на основе одного из параметров. Найти овощи в салате, соответствующие заданному диапазону калорийности.
5. **Звукозапись.** Определить иерархию музыкальных композиций. Записать на диск сборку. Посчитать продолжительность. Провести перестановку композиций диска на основе

принадлежности к стилю. Найти композицию, соответствующую заданному диапазону длины треков.

6. **Камни.** Определить иерархию драгоценных и полудрагоценных камней. Отобрать камни для ожерелья. Посчитать общий вес (в каратах) и стоимость. Провести сортировку камней ожерелья на основе ценности. Найти камни в ожерелье, соответствующие заданному диапазону параметров прозрачности.
7. **Рыцарь.** Определить иерархию амуниции рыцаря. Экипировать рыцаря. Посчитать стоимость. Провести сортировку амуниции на основе веса. Найти элементы амуниции, соответствующие заданному диапазону параметров цены.
8. **Транспорт.** Определить иерархию подвижного состава железнодорожного транспорта. Создать пассажирский поезд. Посчитать общую численность пассажиров и багажа. Провести сортировку вагонов поезда на основе уровня комфортности. Найти вагоны в поезде, соответствующие заданному диапазону параметров числа пассажиров.
9. **Авиакомпания.** Определить иерархию самолетов. Создать авиакомпанию. Посчитать общую вместимость и грузоподъемность. Провести сортировку самолетов компании по дальности полета. Найти самолет в компании, соответствующий заданному диапазону параметров потребления горючего.
10. **Таксопарк.** Определить иерархию легковых автомобилей. Создать таксопарк. Посчитать стоимость автопарка. Провести сортировку автомобилей парка по расходу топлива. Найти автомобиль в компании, соответствующий заданному диапазону параметров скорости.
11. **Страхование.** Определить иерархию страховых обязательств. Собрать из обязательств дериватив. Посчитать стоимость. Провести сортировку обязательств в деривативе на основе уменьшения степени риска. Найти обязательство в деривативе, соответствующее заданному диапазону параметров.
12. **Мобильная связь.** Определить иерархию тарифов мобильной компании. Создать список тарифов компании. Посчитать общую численность клиентов. Провести сортировку тарифов на основе размера абонентской платы. Найти тариф в компании, соответствующий заданному диапазону параметров.
13. **Фургон кофе.** Загрузить фургон определенного объема грузом на определенную сумму из различных сортов кофе, находящихся в тому же в разных физических состояниях (зерно, молотый, растворимый в банках и пакетиках). Учитывать объем кофе вместе с упаковкой. Провести сортировку товаров на основе соотношения цены и веса. Найти товар в фургоне, соответствующий заданному диапазону параметров качества.
14. **Игровая комната.** Подготовить игровую комнату для детей разных возрастных групп. Игрушек должно быть фиксированное количество в пределах выделенной суммы денег. Должны встречаться игрушки родственных групп, н-р: маленькие, средние и большие машины, куклы, мячи, кубики. Провести сортировку игрушек в комнате по одному из параметров. Найти игрушки в комнате, соответствующие заданному диапазону параметров.
15. **Налоги.** Определить множество и сумму налоговых выплат физического лица за год с учетом доходов с основного и дополнительного мест работы, авторских вознаграждений, продажи имущества, получения в подарок денежных сумм и имущества, переводов из-за границы, льгот на детей и материальную помощь. Провести сортировку налогов по сумме.
16. **Вклады.** Сформировать набор предложений клиенту по вкладам различных банков для оптимального выбора. Учитывать возможность досрочного снятия кредита и\или пополнения. Реализовать поиск и сортировку вкладов.
17. **Туристические путевки.** Сформировать набор предложений клиенту по выбору туристической путевки различного типа (отдых, экскурсии, лечение, шопинг, круиз и т д) для оптимального выбора. Учитывать возможность выбора транспорта, питания и числа дней. Реализовать выбор и сортировку путевок.
18. **Кредиты.** Сформировать набор предложений клиенту по целевым кредитам различных банков для оптимального выбора. Учитывать возможность досрочного погашения кредита и\или увеличения кредитной линии. Реализовать выбор и поиск кредита.

## Лабораторное занятие 6-7.

### Интерфейсные объекты

#### Задания

1. Нарисовать картинку, изображающую:

- деревенский дом и окружающую природу (озеро, деревья);
- городской дом и шоссе с автомобилем (или другим транспортом);
- снеговика в зимнем лесу (ели, деревья без листьев);
- море, берег, парусник.

Разработать и реализовать индивидуально подобный проект дома.

2. Разработать программу рисования графических примитивов. Выбор графического примитива осуществить с помощью элемента управления **ListBox**, используя оператор выбора **Case**.

3. Добавить в программу рисования графических примитивов возможность изменения цвета фона формы **BackColor** с помощью перебора стандартных констант цвета. Также предусмотреть функцию выбора цвета для рисуемого графического примитива. Выбор организовать при помощи элементов управления **ListBox**.

## Лабораторное занятие 8-9.

### Графические объекты

#### Задание

1. Используя готовые картинки в формате BMP построить:

- рощу лиственных и хвойных деревьев;
- луг из разноцветных цветов;
- штабель из ящиков и коробок;
- небо в облаках.

Разработать и реализовать индивидуально подобный проект дома.

2. В окружении предметов переднего и заднего планов имитировать движение:

- автомобиля;
- самолёта;
- лодки;
- паровоза.

Разработать и реализовать индивидуально подобный проект дома.

## Лабораторное занятие 10-13.

### Типизированные файлы

### Обработка больших объемов данных

#### Задание

1. Разработать типизированные файлы для хранения данных из приложения лабораторных работ 1-5. Осуществить дихотомический поиск в созданных упорядоченных файлах данных проекта. На основе информации из типизированных файлов проекта создать хэш-файл и осуществить в нем поиск данных.

2. Разработать в проекте методы, добавив их интерфейсному модулю, для реализации алгоритмов внешней сортировки, слияния строго упорядоченных файлов и слияния нестрого упорядоченных файлов.

## 7 семестр

## Лабораторное занятие №1-2.

### Реляционные базы данных

#### Задания.

1. Разработать однотабличную базу данных средствами MS Access, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля,

логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей). Подключить созданную базу данных к проекту C#, вывести содержимое базы данных на форму.

Примеры. 1. БД «Легковые автомобили». 2. БД «Сотовые телефоны». 1. БД «Бытовая техника». 1. БД «Спортсмены».

### **Лабораторное занятие №3-4.**

#### *Проектирование базы данных*

Задания.

1. Разработать однотабличную базу данных средствами C#, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей). Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.

### **Лабораторное занятие №5-6.**

#### *Создание БД, навигация по таблице*

Задания.

1. Разработать двухтабличную базу данных средствами C#, содержащую 3-4 поля разного типа в каждой таблице. Заполнить базу данных произвольной информацией (25-50 записей). Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.

Реализовать в проекте выполнение теоретико-множественных операций: объединение множеств, пересечение множеств, разность и симметрическая разность множеств. Результаты операций вывести в виде таблиц базы данных на экран.

### **Лабораторное занятие №7-8.**

#### *Фильтрация, индексирование и поиск данных*

Задания.

1. Запрограммировать сортировку в разработанных базах данных по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.

2. Разработать запросы к базе данных, созданной средствами C#, с помощью хранимых процедур. Реализовать запросы на добавление, удаление, сортировку и поиск данных.

### **Лабораторное занятие №9-10.**

#### *Введение в язык SQL*

Задания.

1. Разработать многотабличную базу данных средствами C#, содержащую 4-5 таблиц по 3-5 полей в каждой. Заполнить базу данных соответствующей информацией (25-50 записей). Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации. Нормализовать (к первой и второй нормальным формам) и денормализовать разработанную базу данных. Запрограммировать сортировку в разработанных базах данных по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.

### **Лабораторное занятие №11-12.**

#### *Запросы на SQL*

Задания.

1. Разработать многотабличную базу данных средствами C#, содержащую 4-5 таблиц по 3-5 полей в каждой. Заполнить базу данных соответствующей информацией (25-50 записей). Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации. Реализовать простые и сложные запросы к различным типам полей базы данных с помощью языка SQL.

## Лабораторное занятие №13-14.

### Основы защиты информации

Задания.

1. Разработать проект, содержащий главное меню. Пункты меню должны быть названы "Задача 1", ..., "Задача 8". При выборе пункта меню на главной форме проекта должно отражаться условие задачи и располагаться кнопка "Решение". По нажатию на кнопке задача должна запускаться на исполнение. В качестве задач взять по одной задаче из восьми блоков предыдущего семестра.

2. Разработать проект шифрования данных с использованием шифра Цезаря. На форме вводится шифруемое сообщение и задается значение ключа. Нажатие на кнопку "Шифрование" приводит к зашифровке сообщения, а - на кнопке "Дешифрование" к его расшифровке (можно сделать в различных текстовых полях). Сообщение может содержать различные буквы одного из алфавитов - строчные и заглавные. Предусмотреть в проекте шифрование на русском и английском языке - можно с помощью разных полей (в коде программы оформить шифрование в виде различных методов).

## Лабораторное занятие №15-16.

### Методы защиты информации. Шифрование данных

Задания.

1. Разработать проект шифрования данных с использованием шифра Виженера. Предусмотреть шифрование и дешифрование данных для строкового сообщения, записанного на английском языке. На форме вводится шифруемое сообщение и задается значение ключевой фразы. Нажатие на кнопку "Шифрование" приводит к зашифровке сообщения, а - на кнопке "Дешифрование" к его расшифровке (можно сделать в различных текстовых полях).

2. Разработайте метод шифрования 32 русских букв, где

1) соседние буквы одной пары меняются местами

2) меняются местами буквы симметричные относительно середины алфавита

3) код К буквы изменяется операцией (K XOR 21)

4) код К буквы изменяется операцией (NOT K)

Является ли способ шифрования симметричным?

Запрограммируйте шифровку и дешифровку данных.

3. Реализовать проект шифрования данных с использованием иных шифров, например RSA.

Задания для лабораторных работ по дисциплине «Программирование» предоставляется студентам на занятиях в электронном виде.

### Самостоятельная работа

Текущая самостоятельная работа направлена на углубление и закрепление знаний студентов и развитие их практических умений. Она заключается в работе с лекционными материалами, поиске и обзоре литературы и электронных источников, информации по заданным темам курса, опережающей самостоятельной работе, в изучении тем, вынесенных на самостоятельную проработку, подготовке к лабораторным занятиям.

Самостоятельная внеаудиторная работа студентов состоит в:

- проработке лекционного материала, составлении конспекта лекций по темам, вынесенным на самостоятельное изучение;
- выполнении домашних заданий.

### Темы для самостоятельного изучения

1. Организация стека. Задачи на стек.
2. Организация очереди. Задачи на очередь.
3. Формирование однонаправленных списков и примеры их использования.

4. Двухнаправленные списки и примеры их использования.
5. Понятие дерева. Обход дерева.
6. Задача о достижимости вершин.
7. Понятие дерева. Задачи, приводящие к деревьям.
8. Пути и циклы в графе.
9. Использование возможностей SQL при работе с базами данных.
10. Шифрование и дешифрование информации.

## **6. Критерии оценивания результатов освоения дисциплины (модуля)**

### **6.1. Оценочные средства и критерии оценивания для текущей аттестации**

#### **5 семестр**

#### **Теоретические вопросы**

1. Что относят к основным понятиям языка программирования C#? В чем они заключаются?
2. Что такое идентификаторы? Каковы правила их написания?
3. Что относят к зарезервированным словам языка C#?
4. С какого служебного слова начинается программа, написанная на языке C#?
5. Каким символом заканчивается текст программы?
6. Из каких двух основных частей состоит программа?
7. Как запустить программу на языке C# на исполнение?
8. Какие комбинации клавиш служат для быстрого выполнения команд в языке C#?
9. Какое служебное слово используется для описания раздела констант?
10. Каким служебным словом начинается раздел описания переменных?
11. Чем вызвана необходимость использования комментариев в программе?
12. Какими символами ограничивается текст комментария? Где в программе можно написать комментарий?
13. Какое влияние оказывает комментарий на выполнение программы?
14. Какая часть программы, описательная или исполнительная, является обязательной?
15. Верно ли, что в программе на языке C# надо описывать все используемые переменные?
16. Для чего используется указание типа величины?
17. Какие группы типов данных можно выделить в языке программирования C#?
18. Какие типы данных относят в языке C# к стандартным типам?
19. Какие типы данных относят в языке C# к простым типам?
20. Какие значения имеют переменные в начале выполнения программы?
21. Можно ли в процессе выполнения программы изменять значения констант?
22. Какие структурированные операторы используются в языке C#?
23. Назовите процедуры ввода информации. В чем их отличие друг от друга?
24. Сколько аргументов может иметь процедура *ReadLine*? Как они записываются?
25. Сколько процедур *ReadLine* может быть в программе?
26. Какой оператор используется для того, чтобы вводимые данные читались с новой строки?
27. Может ли быть пустым список ввода? В чем назначение такой команды?
28. Надо ли описывать тип переменных, входящих в список ввода?
29. Какое соответствие должно быть между типами переменных списка ввода и типами данных, вводимых с клавиатуры?
30. Каково назначение процедуры *WriteLine*?
31. Может ли элементом списка вывода быть константа?
32. Можно ли с помощью процедуры *WriteLine* напечатать значение выражения?
33. Какие обозначения используются в форматах вывода?
34. В чем состоит действие процедуры *WriteLine* без параметров?
35. Как называются алгоритмы, в которых используется только структура следования?
36. Допустимо ли присваивание переменной целого типа значения выражения вещественного типа?

37. Допустимо ли присваивание переменной вещественного типа значения выражения целого типа?
38. Существуют ли ограничения на множество целых чисел, используемых в языке C#? Если да, то чем они определяются?
39. Каковы основные операции, определенные для данных целочисленного типа?
40. Перечислите основные стандартные функции, определенные для данных целочисленного типа. Каков тип результата, возвращаемого каждой функцией?
41. Для представления какой информации в языке C# используется тип *Double*?
42. В чем отличие двух типов числовых данных *Double* и *Int*?
43. Какие две формы записи вещественных чисел используются в языке C#?
44. В чем сущность понятий «диапазон» и «точность» как важных характеристик вещественных переменных?
45. Перечислите основные стандартные функции, определенные для данных вещественного типа. Каков тип результата, возвращаемого каждой функцией?
46. Что называется мантиссой числа? Что называется порядком числа?
47. Какие стандартные тригонометрические функции, реализованные в языке C#, вы знаете?
48. Почему в языке C# аргумент стандартной функции всегда записывается в скобках?
49. Почему знак умножения всегда выписывается явно?
50. К каким типам числовой информации применимы функции *sqr(x)* и *sqrt(x)*?
51. Для какого типа данных определены операции +, -, \*, /?
52. Какие две операции определены для данных целого типа, но не определены для данных вещественного типа?
53. Какие операции отношения применимы к числовым данным целого и вещественного типа?
54. Каков приоритет выполнения арифметических операций?
55. К данным какого типа применимы функции *trunc(x)* и *round(x)*?
56. Почему запись формул на языке C# линейна?
57. Какая функция вычисляет модуль вещественного числа  $x$ ?
58. Какая функция возвращает натуральный логарифм числа  $x$ ?
59. К каким типам данных применимы арифметические операции / и %?
60. Что делает стандартная функция *trunc(x)*?
61. Каков результат работы стандартной функции *round(x)*?
62. При каких значениях аргумента стандартные функции *trunc(x)* и *round(x)* возвращают одинаковый результат?
63. При каком значении аргумента  $x$  не определен результат работы функции *ln(x)*?
64. Может ли пробел входить в число символов, составляющих идентификатор?
65. Чему равен результат целочисленного деления, если делимое меньше делителя?
66. Какие элементы языка C# записываются с помощью составных специальных символов?
67. Как называются слова, имеющие фиксированное начертание и раз и навсегда определенный смысл?
68. Из каких символов может состоять идентификатор?
69. Как называются элементы данных языка, значения которых установлены в описательной части программы и не изменяются в процессе ее выполнения?
70. Можно ли ключевые слова использовать в качестве имен переменных?
71. Как называются величины, которые могут менять свое значение в процессе выполнения программы?
72. Можно ли при записи арифметических выражений на языке C# использовать фигурные скобки?
73. Можно ли при записи арифметических выражений на языке C# использовать квадратные скобки?
74. Какие функции преобразуют вещественный аргумент в целое число?
75. Что такое «выражение», «операция», «операнд»?

76. Как называется некоторый конечный набор операций (действий, предписаний), выполнение которых одна за другой за конечное число шагов приводит к намеченной цели (искомому результату, решению поставленной задачи)?
77. Можно ли при записи идентификаторов использовать буквы русского алфавита?
78. В каких случаях применяется условный оператор *If ... .. Else ...* в языке программирования C#?
79. Как называются программы, в основе которых лежит структура *следование*?
80. Какие служебные слова могут использоваться при записи условного оператора?
81. Какой тип должно иметь выражение, стоящее в условном операторе между словами *If ...* ?
82. Какое служебное слово используется для описания данных логического типа?
83. Какие значения могут принимать данные логического типа?
84. Почему логический тип относится к порядковым типам?
85. Верно ли, что *true > false*?
86. Как расположить логические операции в порядке убывания их приоритета?
87. Верно ли, что результат операции логического умножения возвращает *true*, если оба операнда истинны?
88. Верно ли, что операция *And* дает *false*, если хотя бы один операнд имеет значение *false*?
89. Верно ли, что результат логического сложения истинен, если хотя бы один операнд имеет значение *true*?
90. Какой тип должна иметь переменная в левой части оператора присваивания, если в его правой части стоит выражение логического типа?
91. Может ли булево выражение содержать круглые скобки?
92. Может ли булево выражение содержать несколько операций?
93. Влияют ли круглые скобки на порядок вычисления значений логических операций в булевых выражениях?
94. Могут ли в булевы выражения входить операции сравнения?
95. Верно ли, что в булевом выражении, содержащем несколько операций, сначала определяются значения всех операций сравнения?
96. Как выглядит односторонняя форма условного оператора?
97. Как выглядит полная форма условного оператора?
98. Могут ли быть операторы *If* вложенными?
99. Как осуществляется вложение условных операторов?
100. Какие операторы входят в состав разветвляющихся программ?
101. Каким образом используются операторские скобки при программировании вложенных условных операторов?
102. Что называется циклом?
103. Как называются программы, в основе которых лежит структура повторения?
104. Каково отличительное свойство итерационного цикла?
105. Сколько операторов итерационного цикла вам известно?
106. В каком случае итерационный цикл называется циклом с предусловием?
107. Каков синтаксис цикла с предусловием в языке C#?
108. Каков тип выражения в операторе цикла с предусловием?
109. Верно ли, что в теле цикла с предусловием должен находиться один оператор?
110. Как сделать, чтобы в теле цикла было несколько операторов?
111. Когда проверяется истинность выражения в цикле *While*?
112. Верно ли, что истинность выражения в цикле с предусловием является условием продолжения цикла?
113. Сколько раз выполнится оператор в теле цикла *While*, если с самого начала значение выражения равно *false*?
114. Верно ли, что цикл *while* используется при вычислении всякого рода сумм и произведений, когда заранее не известно число повторений?
115. В каком случае итерационный цикл называется циклом с постусловием?
116. Каков синтаксис цикла с постусловием в языке C#?



117. Какой вид имеет цикл *Do while* без операторных скобок?
118. Когда проверяется истинность выражения в операторе цикла *Do while*?
119. Какой тип имеет выражение в операторе цикла с постусловием?
120. Почему в цикле *Do while* оператор тела цикла всегда будет выполнен хотя бы один раз?
121. Верно ли, что истинность выражения в цикле *Do while* является условием окончания цикла?
122. В каких случаях целесообразно использовать цикл с предусловием *While*?
123. В каких случаях целесообразно использовать цикл с предусловием *Do while*?
124. Можно ли войти в тело цикла *While*, минуя его заголовок?
125. Можно ли войти в тело цикла *Do while*, минуя его заголовок?
126. Допустим ли выход из тела цикла *While*?
127. Допустим ли выход из тела цикла *Do while*?
128. В чём отличие арифметического цикла от итерационного?
129. В каком случае может использоваться цикл со счётчиком?
130. Какие модификации цикла со счётчиком существуют в языке программирования Turbo Pascal?
131. Что называется счётчиком цикла *for*?
132. Верно ли, что цикл с параметром применяется в тех случаях, когда заранее известно число повторений?
133. С помощью каких служебных слов записывается цикл с параметром?
134. Какой тип должен иметь параметр цикла в цикле *for*?
135. Какой тип должны иметь начальное и конечное значения в цикле *for*?
136. Желательно ли изменять значение управляющей переменной в теле цикла со счётчиком?
137. Чему равно значение управляющей переменной *for* после завершения цикла?
138. Можно ли войти в тело цикла *for*, минуя его заголовок?
139. Допустим ли выход из тела цикла *for*?
140. Какие циклы называются вложенными?
141. Верно ли, что цикл, содержащий в себе другой цикл, называется внешним?
142. Какой цикл называется внутренним?
143. Какому условию должны удовлетворять вложенные циклы?
144. Может ли внешний цикл быть циклом *for*, а внутренний – циклом *Do while*?
145. Могут ли внутренний и внешний циклы быть циклами разных видов?
146. Могут ли внутренний и внешний циклы быть циклами одного вида?
147. Может ли вещественная переменная быть параметром цикла *for*?
148. Можно ли в качестве параметров вложенных циклов *for* использовать одну и ту же переменную?
149. Верно ли, что любой цикл *for* можно записать при помощи цикла *While*?
150. Верно ли, что любой цикл *while* можно записать при помощи цикла *for*?
151. Верно ли, что не любой цикл *for* можно записать при помощи цикла *While*?
152. Верно ли, что не любой цикл *while* можно записать при помощи цикла *for*?
153. Что такое массив?
154. Верно ли, что массив – это структурированный тип данных, состоящий из фиксированного числа элементов, имеющих один и тот же тип?
155. Какими двумя свойствами должен обладать массив?
156. Какие типы данных не допустимы для элементов массива? Почему?
157. В каком разделе (разделах) программы допустимо описание массива?
158. Какой массив называется одномерным?
159. Верно ли, что одномерный массив соответствует понятию линейной таблицы (вектора)?
160. Как задаётся одномерный массив в языке программирования C#?
161. В какой последовательности располагаются в памяти элементы одномерного массива?
162. Верно ли, что элементы с большими значениями индекса хранятся в больших адресах памяти?
163. Где и как определяется общее число элементов массива?

164. Каким образом осуществляется описание типа массива в языке программирования C#?
165. Можно ли в описании массива использовать предварительно определенные константы?
166. Можно ли размерность массива определить с помощью типа диапазон?
167. Какие типы данных можно использовать при описании индекса массива?
168. Может ли левая граница индексов массива быть меньше правой?
169. Может ли левая граница индексов массива быть положительной?
170. Может ли левая граница индексов массива быть равной нулю?
171. Может ли левая граница индексов массива быть отрицательной?
172. Как осуществляется доступ к каждому элементу массива?
173. Когда индекс элемента в массиве совпадает с порядковым номером этого элемента?
174. Может ли индекс быть выражением целого типа?
175. Может ли индекс быть выражением вещественного типа?
176. Может ли индекс быть выражением любого порядкового типа?
177. Всегда ли работа с массивом сводится к работе с его компонентами?
178. Обязательно ли количество элементов массива должно быть фиксированным, то есть определяться при трансляции программы?
179. Какие операции определены над массивами в языке программирования C#?
180. Может ли процедура *ReadLine(x)* ввести весь массив *x* целиком?
181. Можно ли с помощью стандартной процедуры *WriteLine(x)* вывести весь массив *x* целиком?
182. Как осуществить сравнение двух массивов?
183. В чем заключается алгоритм сортировки «пузырьком»?
184. В чем заключается алгоритм сортировки методом простого выбора?
185. В чем заключается алгоритм сортировки методом Шелла?
186. В каком разделе (разделах) программы допустимо описание массива?
187. Какой массив называется двумерным?
188. Где и как определяется общее число элементов массива?
189. Какие типы данных можно использовать при описании индекса массива?
190. Можно ли размерность массива определить с помощью типа диапазон?
191. Каким образом осуществляется описание типа массива в языке программирования C#?
192. Можно ли в описании массива использовать предварительно определенные константы?
193. Как описывается двумерный массив в языке программирования C#?
194. Как организовывается обращение к элементу двумерного массива в языке программирования C#?
195. Могут ли левые границы индексов двумерного массива быть меньше правых?
196. Могут ли левые границы индексов двумерного массива быть положительными?
197. Могут ли левые границы индексов двумерного массива быть равными нулю?
198. Могут ли левые границы индексов двумерного массива быть отрицательными?
199. Может ли индекс в правильно составленной программе выходить за пределы, определенные типом диапазон?
200. Могут ли индексы двумерного массива быть выражениями целого типа?
201. Могут ли индексы двумерного массива быть выражениями вещественного типа?
202. Могут ли индексы двумерного массива быть выражениями любого порядкового типа?
203. Верно ли, что двумерный массив соответствует понятию прямоугольной таблицы (матрице, набору векторов)?
204. Верно ли, что двумерные массивы располагаются в памяти таким образом, что второй индекс изменяется быстрее, чем первый?
205. Верно ли, что матрица расположена в памяти по строкам?
206. Верно ли, что могут существовать массивы массивов?
207. Какой массив называется *n*-мерным?
208. Верно ли, что элементами массива могут быть данные любого типа, включая структурированные?
209. Существуют ли ограничения на размерность массива?

210. Всегда ли работа с массивом сводится к работе с его компонентами?
211. Обязательно ли количество элементов массива должно быть фиксированным, то есть определяться при трансляции программы?
212. Как можно симитировать работу с массивом переменной длины?
213. Какие существуют способы вывода элементов двумерного массива?
214. Верно ли, что литерный (символьный) тип относится к скалярным типам данных?
215. Как задается описание переменных литерного типа?
216. Каково множество значений литерного типа?
217. Какой объем памяти требуется для хранения переменной символьного типа?
218. Верно ли, что значением литерного типа является множество всех символов кодовой таблицы ПЭВМ?
219. Что такое код символа?
220. Верно ли, что каждому символу ставится в соответствие целое число в диапазоне 0 .. 255?
221. Можно ли к данным символьного типа применять операции отношения?
222. Верно ли, что в программе значения символьного типа должны быть заключены в апострофы?
223. Каковы особенности кодировки цифр, латинских и русских букв?
224. Какие операции применимы к символьным данным?
225. Какие встроенные функции можно применить к символьным данным?
226. Что называется строкой в языке программирования Turbo Pascal?
227. Является ли тип *string* структурированным типом данных?
228. Что представляет собой тип *string*?
229. Какое максимально возможное количество символов может содержать строка?
230. Каково функциональное назначение типа *string*?
231. Как при описании строкового типа указывается длина строки?
232. Всегда ли фактическая длина строки равна объявленной в описании программы?
233. Какие типовые операции над строками определены в языке программирования С#?
234. Каким образом в переменную типа *string* можно поместить какое-то конкретное значение?
235. Может ли в процессе выполнения программы изменяться фактическая длина строки?
236. Может ли в процессе выполнения программы фактическая длина строки стать больше, чем объявлено в описании? Что произойдет в этом случае?
237. Если длина строки не указана при объявлении типа *string*, то чему она равна по умолчанию?
238. Могут ли переменные строкового типа быть операндами выражений?
239. Верно ли, что тип *string* аналогичен одномерному массиву символов *array [0..n] of char*?
240. Чем отличается тип *string* от одномерного массива символов *array [0..n] of char*?
241. Верно ли, что к любому символу в строке можно обратиться точно так же, как к элементу одномерного массива?
242. Какие операции отношения можно применять к переменным типа *string*? Каков их приоритет?
243. Как происходит сравнение строк одинаковой длины?
244. Как происходит сравнение строк разной длины?
245. Какие основные процедуры и функции для обработки строк определены в языке программирования С#?
246. Как работает функция *Length*?
247. Как работает функция + и какой операцией её можно заменить?
248. Как работает функция *Copy*?
249. Как работает процедура *Delete*?
250. Как работает процедура *Insert*?
251. Какое значение возвращает функция *IndexOf* и каково значение её аргументов?
252. Как работает функция *CompareTo*?

### Критерии оценивания теоретических вопросов

#### 1. Нормы оценивания ответов на теоретические вопросы

№ п/п	Теоретический вопрос	Количество баллов (*)
1	Дан краткий ответ на поставленный вопрос	1 балл
2	Дан развернутый ответ на вопрос с анализом результатов	2 балла

(\*) Возможна градация в 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за ответы на теоретические вопросы выставляется, если набрано не менее 3 баллов при ответе на три вопроса, в противном случае выставляется «не зачтено».

### Задания для лабораторных занятий

Задачи по темам курса предложены к каждому лабораторному занятию.

Задания для лабораторных и самостоятельной работ, образцы решений основных типовых задач практики также размещены в системе дистанционного обучения СмолГУ ([www.moodle.smolgu.ru](http://www.moodle.smolgu.ru)).

### Критерии оценивания выполнения лабораторных работ

#### 1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(\*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

## 6 семестр

### Проверочная работа

1. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
2. Дано трехзначное число. Определить, какая из его цифр больше: первая или последняя.
3. Найти сумму значений функции  $f(n)=x^2+12$  по всем целым значениям из диапазона от 1 до 40 с помощью цикла с предусловием, с постусловием и цикла с параметром.
4. Найти сумму элементов одномерного массива, имеющих четные индексы.
5. В данном двумерном массиве найти все элементы, являющиеся наибольшими среди 4 своих непосредственных соседей (сверху, снизу, слева и справа).
6. Заменить в заданной строке последнее вхождение слова «вечер» словом «утро».

### Критерии оценивания проверочной работы

#### 1. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(\*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

### Задания для лабораторных занятий

Задачи по темам курса предложены к каждому лабораторному занятию.

Задания для лабораторных и самостоятельной работ, образцы решений основных типовых задач практики также размещены в системе дистанционного обучения СмолГУ ([www.moodle.smolgu.ru](http://www.moodle.smolgu.ru)).

### Критерии оценивания выполнения лабораторных работ

1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(\*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

## 7 семестр

### Задания для лабораторных занятий

Задачи по темам курса предложены к каждому лабораторному занятию.

Задания для лабораторных и самостоятельной работ, образцы решений основных типовых задач практики также размещены в системе дистанционного обучения СмолГУ ([www.moodle.smolgu.ru](http://www.moodle.smolgu.ru)).

### Критерии оценивания выполнения лабораторных работ

1. Нормы оценивания каждой лабораторной работы:

№п/п	Структурная часть работы	Количество баллов (*)
1	Ответ на теоретические вопросы по теме лабораторной работы	1 балл
2	Демонстрация выполнения конкретного задания, предложенного для самостоятельного решения к лабораторной работе	2 балла

(\*) с возможностью градации до 0,25 балла.

2. Шкала оценивания. Оценка «зачтено» за лабораторную работу выставляется, если набрано не менее 2 баллов, в противном случае за работу выставляется «не зачтено».

## 6.2. Оценочные средства и критерии оценивания для промежуточной аттестации

### 5 семестр

#### Зачетная контрольная работа 1

1. Вычислить значение выражения:  $\frac{1 + \sin \sqrt{x+1}}{\cos(12y - 4x)} - |x^2 + y^3|$ .

2. Дано трехзначное число. Определить, является ли произведение его цифр трехзначным числом.

3. Дано натуральное число  $n$ . Вычислить сумму вида  $s = \frac{2}{1} + \frac{3}{2} + \frac{4}{3} + \dots + \frac{n+1}{n}$  с помощью цикла с предусловием.

4. Вывести на экран таблицу значений аргумента  $x$  и функции  $f(x) = 2 \cdot \sin^2 x + 3x$  на отрезке  $[a, b]$  с шагом  $h$ .

5. Дан одномерный массив целых чисел. Поменять местами первый положительный и последний наименьший элементы массива.

6. Найти в данном двумерном массиве размера  $n \times m$  количество и сумму элементов, кратных заданному числу  $a$ , и вывести их местоположение.

7. Заменить в заданной строке первое вхождение слова «неправильно» словом «правильно».

### Критерии оценивания зачетной контрольной работы 1

#### 1. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(\*) Возможна градация в 0,25 балла.

#### 2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

### Зачетная контрольная работа 2

1. Сформировать стек из  $N$  натуральных чисел, просмотреть его и увеличить все четные его элементы на 1.

2. Сформировать очередь из  $N$  натуральных чисел, просмотреть ее и найти произведение второго и пятого элементов очереди. Результат поместить в очередь.

В данных задачах разработать собственные методы инициализации, помещения и извлечения элемента и просмотра стека и очереди.

3. На клеточной карте заданы острова (-1) и водная поверхность. Вычислить количество островов и площадь каждого острова.

### Критерии оценивания зачетной контрольной работы 2

#### 3. Нормы оценивания работы

№ п/п	Структурная часть контрольной работы	Количество баллов (*)
1	Правильно реализован каждый метод решения	1 балл
2	Анализ результатов	2 балла

(\*) Возможна градация в 0,25 балла.

#### 4. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

## Критерий получения зачета

Зачет выставляется по результатам работы студента в течение семестра согласно Положению о текущем контроле успеваемости и промежуточной аттестации обучающихся в федеральном государственном бюджетном образовательном учреждении высшего профессионального образования «Смоленский государственный университет» (утверждено приказом ректора № 01-113 от 26.09.2019 г.; внесены дополнения приказом ректора № 01-48 от 30.04.2020).

Для получения зачета студент должен:

- уметь отвечать на теоретические вопросы, рассмотренные на лекциях;
- уметь решать задачи, предложенные на лабораторных занятиях;
- уметь решать задачи, предложенные на зачетной контрольной работе.

## 6 семестр

### Вопросы к экзамену

1. Среда программирования C#: описание основных окон.
2. Основные компоненты среды программирования C#.
3. Событие и процедура обработки события. Подсказки, навигация, справка.
4. Структура проекта в среде программирования C#.
5. Раздел описаний и операторов. Простые типы данных.
6. Арифметические выражения. Присваивание. Ввод и вывод данных.
7. Линейные программы. Обработка целочисленных данных.
8. Логические выражения. Ветвление, безусловный переход.
9. Циклы с условиями и с параметром.
10. Общие сведения о формах языка программирования C#.
11. Основные элементы управления на форме в среде программирования C#.
12. Организация меню и списков на форме в среде программирования C#.
13. Использование флажков и переключателей в языке программирования C#.
14. Объявление, обработка и вывод массива в среде программирования C#.
15. Использование компонента TextBox для вывода одномерного массива.
16. Использование компонента DataGridView для вывода одномерного массива.
17. Сортировка данных в одномерном массиве методом «пузырька».
18. Сортировка данных в одномерном массиве методом обмена.
19. Сортировка данных в одномерном массиве методом вставки.
20. «Быстрые» сортировки данных в одномерном массиве.
21. Использование компонента DataGridView для вывода двумерного массива.
22. Сортировка данных в двумерном массиве методом «пузырька».
23. Сортировка данных в двумерном массиве методом обмена.
24. Сортировка данных в двумерном массиве методом вставки.
25. Обработка строк в среде программирования C#.
26. Основные способы организации методов в среде программирования C#.
27. Формальные и фактические параметры, локальные и глобальные переменные в подпрограммах.
28. Модификатор Params, перегрузка методов, рекурсивные методы среды C#.
29. Структуры данных, коллекции и классы прототипы языка C#.
30. Основные методы работы со стеком и очередью в среде C#.
31. Организация стека с помощью массива в среде C#..
32. Организация очереди с помощью массива в среде C#..
33. Общие сведения о графике и GDI+ языка программирования C#.
34. Основные пространства имен GDI+ языка программирования C#.
35. Работа с основными типами System.Drawing языка программирования C#.
36. Работа с цветом и шрифтами в среде программирования C#.
37. Работа с перьями и кистями в среде программирования C#.

38. Организация движения в среде программирования C#.
39. Использование сплайнов при программировании компьютерной графики.
40. Программирование фракталов в среде программирования C#.
41. Основные сведения о графической библиотеке OpenGL.
42. Рисование геометрических объектов в среде OpenGL.
43. Кодирование и декодирование данных в среде программирования C#.

### Задачи к экзамену

1. Дано натуральное число  $n$ . Вычислить сумму вида  $s = \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \dots + \frac{1}{(2n+1)^2}$ .
2. Дано натуральное число  $n$ . Вычислить произведение вида  $p = \frac{2}{3} \cdot \frac{3}{4} \cdot \frac{4}{5} \cdot \dots \cdot \frac{n}{n+1}$ .
3. Дано натуральное число  $n$ . Вычислить сумму вида  $s = 1 - \frac{1}{2} + \frac{1}{4} - \frac{1}{8} + \dots + (-1)^n \cdot \frac{1}{2^n}$ .
4. Дано натуральное число  $n$ . Определить количество цифр этого числа.
5. Дано натуральное число  $n$ . Найти сумму первой и последней цифры этого числа.
6. Дано натуральное число  $n$ . Переставить местами первую и последнюю цифры этого числа.
7. Найти количество положительных и отрицательных элементов одномерного массива.
8. Вычислить сумму элементов данного одномерного массива, кратных заданному числу  $k$ .
9. Найти минимальный элемент одномерного массива целых чисел.
10. Найти номер последнего максимального элемента массива целых чисел.
11. Найти количество нечетных элементов одномерного массива целых чисел.
12. Найти сумму элементов одномерного массива, имеющих четные индексы.
13. Дан одномерный массив целых чисел. Найти количество элементов массива, которые положительны и не превосходят заданное число  $a$ .
14. Дан одномерный массив целых чисел. Все элементы с нечетными индексами разделить на первый элемент.
15. Дан одномерный массив целых чисел. Выяснить, является ли сумма элементов массива четным числом.
16. Дан одномерный массив целых чисел. Поменять местами первый наибольший и последний наименьший элементы массива.
17. Дан одномерный массив целых чисел. Переставить местами первый положительный и последний отрицательный элементы массива.
18. Подсчитать количество повторов элементов в данном одномерном числовом массиве.
19. Подсчитать количество различных повторяющихся элементов в данном одномерном числовом массиве.
20. Дан массив чисел. Найти сумму трех минимальных элементов.
21. Отсортировать элементы данного одномерного числового массива в порядке возрастания «пузырьком».
22. Отсортировать элементы данного одномерного числового массива в порядке возрастания методом простого выбора.
23. Отсортировать элементы данного одномерного числового массива в порядке убывания методом вставки.
24. Отсортировать элементы данного одномерного числового массива в порядке убывания методом Шелла.
25. Отсортировать элементы данного одномерного числового массива, стоящие на местах с четными номерами, в порядке возрастания, стоящие на местах с нечетными номерами – в порядке убывания.
26. Дан одномерный массив из 15 целых чисел. Упорядочить первые пять элементов по возрастанию, вторые пять – по убыванию, последние – по невозрастанию с помощью сортировки «пузырьком», метода простого выбора и сортировки Шелла соответственно.



27. Дан одномерный массив целых чисел. Определить значение второго по величине элемента массива.
28. Изменить последовательность значений элементов данного одномерного числового массива на обратный порядок.
29. Изменить последовательность значений элементов данного одномерного числового массива, стоящих на местах с чётными номерами, на обратный порядок (остальные элементы должны остаться на своих местах).
30. Определить, является ли данный одномерный числовой массив упорядоченным по возрастанию.
31. Определить, является ли данный одномерный числовой массив упорядоченным по убыванию.
32. В данном одномерном числовом массиве найти самую длинную неубывающую числовую последовательность и вывести её на экран.
33. Дан упорядоченный по убыванию массив чисел. Ввести число  $N$ . Вставить это число в упорядоченный массив так, чтобы массив-результат тоже был упорядочен по убыванию.
34. Определить, есть ли среди элементов заданного двумерного массива размера  $n \times m$  данное число  $q$ .
35. Дан массив целых чисел  $n \times m$ . Для каждой строки массива определить, имеются ли в ней положительные элементы.
36. Дан массив целых чисел  $n \times m$ . Для каждого столбца массива определить, имеются ли в нем элементы, большие числа  $q$ .
37. Найти в данном двумерном массиве размера  $n \times m$  количество отрицательных и положительных чисел.
38. Найти в данном двумерном массиве размера  $n \times m$  сумму нечетных элементов.
39. Найти в данном двумерном массиве размера  $n \times m$  количество и сумму элементов, кратных заданному числу  $a$ .
40. Выяснить, есть ли одинаковые числа в данном двумерном числовом массиве размера  $n \times m$ .
41. Найти в данном двумерном массиве размера  $n \times n$  наибольший элемент на главной диагонали и вывести его на экран.
42. Установить, имеются ли в данном двумерном массиве элементы, наименьшие в строке и наибольшие в столбце, если такие элементы есть, вывести их на экран.
43. Установить, имеются ли в данном двумерном массиве элементы, наибольшие в обеих своих диагоналях, если такие элементы есть, вывести их на экран.
44. Установить, имеются ли в данном двумерном массиве элементы, наибольшие ровно в одной из двух своих диагоналей, если такие элементы есть, вывести их на экран.
45. Подсчитать количество различных повторяющихся элементов в двумерном числовом массиве.
46. Подсчитать количество повторов элементов в двумерном числовом массиве.
47. Подсчитать количество уникальных элементов в двумерном числовом массиве.
48. Найти в строках данного двумерного массива размера  $n \times m$  самые правые наибольшие элементы и определить их местоположение.
49. Дан двумерный массив целых чисел размера  $n \times m$ . Определить, сколько в нем пар соседних одинаковых элементов. Элементы считаются соседними, если их индексы в столбцах и/или строках различаются не более чем на единицу.
50. В данном двумерном массиве найти все элементы, являющиеся наибольшими среди 4 своих непосредственных соседей (сверху, снизу, слева и справа).
51. В данном двумерном массиве найти все элементы, являющиеся наименьшими среди 8 своих непосредственных соседей.
52. В данном двумерном массиве размера  $n \times n$  вычислить сумму и определить число положительных элементов, находящихся над главной диагональю.
53. Дан массив целых чисел  $n \times m$ . Для каждой строки массива определить, упорядочены ли ее элементы по возрастанию.

54. Определить, есть ли в данном двумерном массиве размера  $n \times m$  элемент, равный нулю, сколько раз он встречается, и вывести его местоположение.
55. В данном двумерном массиве размера  $n \times n$  отсортировать по возрастанию элементы, находящиеся на главной диагонали.
56. В данном двумерном массиве размера  $n \times m$  поменять местами вторую и предпоследнюю строки.
57. В данном двумерном массиве размера  $n \times m$  поменять местами первый и последний столбец.
58. Повернуть двумерный массив размера  $n \times n$  на 90 градусов по часовой стрелке и вывести результат на экран.
59. Определить, сколько раз встречается символ «а» в русских словах заданной строки.
60. Заменить в заданной строке последнее вхождение слова «вечер» словом «утро».
61. Заменить в заданной строке все вхождения слова «неправильно» словом «правильно».
62. Заменить в заданной строке чётные вхождения слова «зима» словом «осень», нечётные – словом «весна».
63. Дана символьная строка. Определить длину самой длинной подстроки из подряд стоящих букв «а».
64. Дана строка символов. Определить, сколько в ней слов, начинающихся и кончающихся на одну и ту же букву.
65. Определить, сколько раз встречается символ «а» в русских словах заданной строки.
66. Заменить в заданной строке последнее вхождение слова «вечер» словом «утро».
67. Заменить в заданной строке все вхождения слова «неправильно» словом «правильно».
68. Заменить в заданной строке чётные вхождения слова «зима» словом «осень», нечётные – словом «весна».
69. Дана символьная строка. Определить длину самой длинной подстроки из подряд стоящих букв «а».
70. Дана строка символов. Определить, сколько в ней слов, начинающихся и кончающихся на одну и ту же букву.
71. Дана строка символов. Дано слово. Удалить из строки все вхождения этого слова.
72. Дана строка символов. Удалить из нее каждое слово нечетной длины. Слова отделяются друг от друга одним пробелом.
73. Дана строка символов. Определить длину самого длинного и самого короткого слова.
74. Дана строка, содержащая произвольный текст. Выяснить, чего в нем больше: русских букв или цифр.
75. Дана строка символов, содержащая текст, заканчивающийся точкой. Вывести на экран слова, содержащие три буквы.
76. Дана строка символов, содержащая текст, заканчивающийся точкой. Вывести на экран те слова, которые содержат хотя бы одну букву s.
77. Дана строка символов, содержащая текст, заканчивающийся точкой. Найти слово, содержащее наибольшее количество указанных букв.
78. Дана строка символов. Группы символов в ней между группами пробелов считаются словами. Определить, сколько слов содержит данная строка.
79. Удалить из строки все повторные вхождения данного (заданного пользователем) символа.
80. Удалить из строки все повторные вхождения символов.

### Образец экзаменационного билета

1. Среда программирования C#: описание основных окон.
2. Сортировка данных в двумерном массиве методом «пузырька».
3. Задача.

### Критерии оценивания ответа на экзамене

1. Нормы оценивания ответа

№п/п	Структурная часть билета	Количество баллов
1	Теоретический вопрос	2 балла
2	Математическая модель	1 балл
3	Реализация решения задачи	2 балла

(\*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

### 7 семестр

#### Вопросы к экзамену

1. Архитектуры СУБД и приложений.
2. Уровни доступа к базам данных.
3. Методологии проектирования приложений баз данных.
4. Использование инструментов Visual Studio и языка C# для доступа к базам данных.
5. Основы современных пользовательских интерфейсов баз данных.
6. Безопасность баз данных.
7. Организация баз и средства управления данными в MS SQL Server.
8. Транзакции и согласованность.
9. Аналитическая обработка данных.
10. Интеллектуальный анализ данных (Data Mining).

#### Задачи к экзамену

1. Разработать однотабличную базу данных «Легковые автомобили» средствами MS Access, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Подключить созданную базу данных к проекту C#, вывести содержимое базы данных на форму.
2. Разработать однотабличную базу данных «Сотовые телефоны» средствами MS Access, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Подключить созданную базу данных к проекту C#, вывести содержимое базы данных на форму.
3. Разработать однотабличную базу данных «Бытовая техника» средствами MS Access, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Подключить созданную базу данных к проекту C#, вывести содержимое базы данных на форму.
4. Разработать однотабличную базу данных «Спортсмены» средствами MS Access, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Подключить созданную базу данных к проекту C#, вывести содержимое базы данных на форму.
5. Разработать однотабличную базу данных «Легковые автомобили» средствами C#, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное

числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.

6. Разработать однотабличную базу данных «Сотовые телефоны» средствами C#, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
7. Разработать однотабличную базу данных «Бытовая техника» средствами C#, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
8. Разработать однотабличную базу данных «Спортсмены» средствами C#, содержащую 5-7 полей разного типа: целочисленное числовое поле, вещественное числовое поле, текстовые поля, логическое поле, поле «дата-время». Заполнить базу данных соответствующей информацией (50-100 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
9. Разработать двухтабличную базу данных «Легковые автомобили» средствами C#, содержащую 3-4 поля разного типа в каждой таблице. Заполнить базу данных произвольной информацией (25-50 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
10. Разработать двухтабличную базу данных «Сотовые телефоны» средствами C#, содержащую 3-4 поля разного типа в каждой таблице. Заполнить базу данных произвольной информацией (25-50 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
11. Разработать двухтабличную базу данных «Бытовая техника» средствами C#, содержащую 3-4 поля разного типа в каждой таблице. Заполнить базу данных произвольной информацией (25-50 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
12. Разработать двухтабличную базу данных «Спортсмены» средствами C#, содержащую 3-4 поля разного типа в каждой таблице. Заполнить базу данных произвольной информацией (25-50 записей), сгенерировав записи случайным образом. Вывести содержимое базы данных на форму, организовать управление данными с помощью стандартной панели навигации.
13. Запрограммировать сортировку в разработанной базе данных «Легковые автомобили» по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.
14. Разработать запросы к базе данных «Легковые автомобили», созданной средствами C#, с помощью хранимых процедур. Реализовать запросы на добавление, удаление, сортировку и поиск данных.
15. Запрограммировать сортировку в разработанной базе данных «Сотовые телефоны» по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.

16. Разработать запросы к базе данных «Сотовые телефоны», созданной средствами С#, с помощью хранимых процедур. Реализовать запросы на добавление, удаление, сортировку и поиск данных.
17. Запрограммировать сортировку в разработанной базе данных «Бытовая техника» по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.
18. Разработать запросы к базе данных «Бытовая техника», созданной средствами С#, с помощью хранимых процедур. Реализовать запросы на добавление, удаление, сортировку и поиск данных.
19. Запрограммировать сортировку в разработанной базе данных «Спортсмены» по столбцам таблицы. Разработать методы фильтрации данных и поиска данных по различным ключам.
20. Разработать запросы к базе данных «Спортсмены», созданной средствами С#, с помощью хранимых процедур. Реализовать запросы на добавление, удаление, сортировку и поиск данных.
21. Разработать проект шифрования данных с использованием шифра Цезаря. Предусмотреть шифрование и дешифрование данных для строкового сообщения, записанного на русском и английском языке.
22. Разработать проект шифрования данных с использованием шифра Виженера. Предусмотреть шифрование и дешифрование данных для строкового сообщения, записанного на русском и английском языке.
23. Разработать проект шифрования данных с использованием шифра RSA. Предусмотреть шифрование и дешифрование данных для строкового сообщения, записанного на русском и английском языке.

### Образец экзаменационного билета

1. Уровни доступа к базам данных.
2. Безопасность баз данных.
3. Задача.

### Критерии оценивания ответа на экзамене

1. Нормы оценивания ответа

№п/п	Структурная часть билета	Количество баллов
1	Теоретический вопрос	2 балла
2	Математическая модель	1 балл
3	Реализация решения задачи	2 балла

(\*) Возможна градация в 0,25 балла.

2. Шкала оценивания работы:

п/п	Оценка	Количество баллов
1	Отлично	4,75-5
2	Хорошо	3,75-4,5
3	Удовлетворительно	3-3,5
4	Неудовлетворительно	менее 3

## 7. Перечень основной и дополнительной учебной литературы

### 7.1. Основная литература

1. Гниденко И. Г. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – Москва: Издательство Юрайт, 2020. – 235 с. – (Высшее образование). – ISBN 978-5-534-02816-4. – URL: <https://urait.ru/bcode/450999>
2. Зимин В. П. Информатика. Лабораторный практикум в 2 ч. Часть 1: учебное пособие для вузов / В. П. Зимин. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 124 с. – (Высшее образование). – ISBN 978-5-534-11588-8. – URL: <https://urait.ru/bcode/451451>
3. Зимин В. П. Информатика. Лабораторный практикум в 2 ч. Часть 2: учебное пособие для вузов / В. П. Зимин. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 153 с. – (Высшее образование). – ISBN 978-5-534-11590-1. – URL: <https://urait.ru/bcode/453949>

4. Зыков С. В. Программирование: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2020. – 320 с. – (Высшее образование). – ISBN 978-5-534-02444-9. – URL: <https://urait.ru/bcode/450832>
5. Крупский В. Н. Теория алгоритмов. Введение в сложность вычислений: учебное пособие для вузов / В. Н. Крупский. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 117 с. – (Высшее образование). – ISBN 978-5-534-04817-9. – URL: <https://urait.ru/bcode/454121>
6. Кувшинов Д. Р. Основы программирования: учебное пособие для вузов / Д. Р. Кувшинов. – Москва: Издательство Юрайт, 2020. – 104 с. – (Высшее образование). – ISBN 978-5-534-07559-5. – URL: <https://urait.ru/bcode/454667>
7. Трофимов В. В. Алгоритмизация и программирование: учебник для вузов / В. В. Трофимов, Т. А. Павловская; под редакцией В. В. Трофимова. – Москва: Издательство Юрайт, 2020. – 137 с. – (Высшее образование). – ISBN 978-5-534-07834-3. – URL: <https://urait.ru/bcode/452333>
8. Черпаков И. В. Основы программирования: учебник и практикум для вузов / И. В. Черпаков. – Москва: Издательство Юрайт, 2020. – 219 с. – (Высшее образование). – ISBN 978-5-9916-9983-9. – URL: <https://urait.ru/bcode/450823>

## **7.2. Дополнительная литература**

1. Гаврилов М. В. Информатика и информационные технологии: учебник для вузов / М. В. Гаврилов, В. А. Климов. – 4-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2020. – 383 с. – (Высшее образование). – ISBN 978-5-534-00814-2. – URL: <https://urait.ru/bcode/449779>
2. Демин А. Ю. Информатика. Лабораторный практикум: учебное пособие для вузов / А. Ю. Демин, В. А. Дорофеев. – Москва: Издательство Юрайт, 2020. – 131 с. – (Высшее образование). – ISBN 978-5-534-08366-8. – URL: <https://urait.ru/bcode/451395>
3. Зыков С. В. Программирование. Объектно-ориентированный подход: учебник и практикум для вузов / С. В. Зыков. – Москва: Издательство Юрайт, 2020. – 155 с. – (Высшее образование). – ISBN 978-5-534-00850-0. – URL : <https://urait.ru/bcode/451488>
4. Казанский А. А. Программирование на Visual C#: учебное пособие для вузов / А. А. Казанский. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2020. – 192 с. – (Высшее образование). – ISBN 978-5-534-12338-8. – URL: <https://urait.ru/bcode/451467>
5. Методы оптимизации: теория и алгоритмы: учебное пособие для вузов / А. А. Черняк, Ж. А. Черняк, Ю. М. Метельский, С. А. Богданович. – 2-е изд., испр. и доп. – Москва: Издательство Юрайт, 2020. – 357 с. – (Высшее образование). – ISBN 978-5-534-04103-3. – URL: <https://urait.ru/bcode/453567>
6. Тузовский А. Ф. Объектно-ориентированное программирование: учебное пособие для вузов / А. Ф. Тузовский. – Москва: Издательство Юрайт, 2020. – 206 с. – (Высшее образование). – ISBN 978-5-534-00849-4. – URL: <https://urait.ru/bcode/451429>
7. Советов Б. Я. Информационные технологии: учебник для вузов / Б. Я. Советов, В. В. Цехановский. – 7-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2020. – 327 с. – (Высшее образование). – ISBN 978-5-534-00048-1. – URL: <https://urait.ru/bcode/449939>

## **7.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»**

1. Система дистанционного обучения СмолГУ ([moodle.smolgu.ru](http://moodle.smolgu.ru)).
2. Национальный открытый университет ([intuit.ru](http://intuit.ru)).
3. Национальная платформа открытого образования ([opened.ru](http://opened.ru))

## **8. Материально-техническое обеспечение**

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, обеспечивающие тематические иллюстрации, соответствующие программе дисциплины (модулей), учебная ауд. 224 на 12 посадочных мест.

Перечень материально-технического обеспечения, необходимого для реализации курса, включает в себя лабораторию, оснащенную персональными компьютерами, объединенные в сеть с выходом в Интернет, проектором и интерактивной доской, ауд.224 на 12 посадочных мест и 6 парт (12 посадочных мест).

Помещение для самостоятельной работы обучающихся оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета, ауд.224 на 12 посадочных мест и 6 парт (12 посадочных мест).

### **9. Программное обеспечение**

1. Операционная система MS Windows XP, Linux.
2. Система программирования MS Visual Studio 19 (язык программирования C#).
3. Поисковые системы сети Интернет.

ДОКУМЕНТ ПОДПИСАН  
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 03B6A3C600B7ADA9B742A1E041DE7D81B0  
Владелец: Артеменков Михаил Николаевич  
Действителен: с 04.10.2021 до 07.10.2022