

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет информатики и информационных технологий

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Кафедра Информационных систем и технологии программирования

Образовательная программа бакалавриата

09.03.02 «Информационные системы и технологии»

Направленность (профиль) программы:

Технологии разработки безопасного программного обеспечения информационных
систем

Уровень высшего образования

Бакалавриат

Форма обучения

очная

Статус дисциплины: входит в обязательную часть ОПОП

Махачкала, 2022

Рабочая программа дисциплины «Технологии программирования» составлена в 2022 году в соответствии с требованиями ФГОС ВО - бакалавриат по направлению подготовки 09.03.02 «Информационные системы и технологии 19.09.2017 № 926.

Разработчик: кафедра информационных систем и технологий программирования, Магомедова С.Р.

Рабочая программа дисциплины одобрена:
на заседании кафедры ИСиТП от «01» марта 2022г., протокол № 8

Зав. кафедрой  Исмиханов З.Н.
(подпись)

на заседании Методической комиссии факультета ИиИТ
от «17» марта 2022г., протокол № 7

Председатель  Бакмаев А.Ш.
(подпись)

Рабочая программа дисциплины согласована с учебно-методическим управлением
«31» марта 2022г.

Начальник УМУ  Гасангаджиева А.Г.

Аннотация рабочей программы дисциплины

Дисциплина «Технологии программирования» входит в обязательную часть ОПОП программы бакалавриата по направлению 09.03.02 «Информационные системы и технологии».

Дисциплина реализуется на факультете Информатики и ИТ кафедрой Информационных систем и технологий программирования.

Содержание дисциплины охватывает круг вопросов, связанных с изучением современных технологий и методов программирования, основных принципов объектно-ориентированного программирования, механизмов доступа к базам данных и работы с ними, приобретением практических навыков использования современных инструментальных средств для разработки, отладки и тестирования создаваемых прикладных программ.

Дисциплина нацелена на формирование следующих компетенций выпускника: общекультурных – ОПК-5, профессиональных – ПК-5, ПК-6.

Преподавание дисциплины предусматривает проведение следующих видов учебных занятий: *лекции, лабораторные занятия, самостоятельная работа.*

Рабочая программа дисциплины предусматривает проведение следующих видов контроля успеваемости в форме *устного и письменного опроса* и промежуточный контроль в форме зачета и экзамена.

Объем дисциплины 8 зачетных единиц, в том числе в академических часах по видам учебных занятий.

Объем дисциплины

Семестр	Учебные занятия						СРС, в том числе экзамен	Форма промежуточной аттестации (зачет, дифференциро- ванный зачет, экзамен
	в том числе							
	Контактная работа обучающихся с преподавателем							
	Всего	Всего	из них					
Лекц ии			Лаборатор ные занятия	Практич еские занятия	КПР			
3	144	68	34	34			76	Зачет
4	144	68	34	34			76	экзамен

1. Цели освоения дисциплины:

Подготовка к самостоятельной профессиональной работе, ознакомление с методами и технологиями программирования, умение ориентироваться во всем многообразии технологий программирования, умение применять практические навыки использования инструментальных и прикладных технологий в различных отраслях техники, экономики, управления и бизнеса.

2. Место дисциплины в структуре образовательной программы:

Дисциплина относится к обязательной части ОПОП учебного плана образовательной программы 09.03.02 «Информационные системы и технологии», профиль подготовки «Общий профиль», изучается в 3, 4 семестре.

Для изучения данной учебной дисциплины необходимы следующие знания, умения и навыки, формируемые предшествующими дисциплинами:

из курса «Программирование на языке высокого уровня»:

Знания: ядро языка программирования высокого уровня, его синтаксис и семантику; основы проектирования программ: типовые алгоритмы.

Умения: описывать разработанные программы посредством блок-схем, тестировать и отлаживать разработанные программы; реализовывать на языке программирования высокого уровня типовые алгоритмы: табуляцию функций, формирование таблиц, нахождение сумм, среднего и т.п.; поиск экстремума, работу с датчиком случайных чисел, ввод и вывод одномерных и двумерных массивов, поиск элементов в массиве, обработку массивов с выводом таблиц, сортировку, ввод и вывод текстов, сравнение фрагментов текста, изменение фрагмента текста по определенному правилу, запись информации в файл, чтение информации из файла, поиск и изменение информации в файле по заданному условию.

Владения: приемами работы в среде программирования (составление, отладка и тестирование программ; разработка и использование интерфейсных объектов)

Перечень последующих учебных дисциплин, для которых необходимы знания, умения и владения, формируемые данной учебной дисциплиной:

- Web-программирование;
- Объектно-ориентированное программирование
- Методы и средства проектирования информационных систем и технологий.

3. Компетенции обучающегося, формируемые в результате освоения дисциплины(перечень планируемых результатов обучения)

Код и наименование компетенции из ОПОП	Код и наименование индикатора достижения компетенции (в соответствии с ОПОП)	Планируемые результаты обучения	Процедура освоения
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	ИД1.ОПК-5.1 .Знает основы системного администрирования, администрирования СУБД, современные стандарты информационного взаимодействия систем	Знает: методы и средства защиты информации в процессе хранения и передачи по компьютерным сетям: классификация, функции	Устный опрос, письменный опрос
	ИД2.ОПК-5.2 Умеет выполнять параметрическую настройку информационных и автоматизированных систем	Умеет: выбирать инструментальные средства и методы разработки программных продуктов.	
	ИД3.ОПК-5.3. Имеет навыки установки программного и	Владеет: методами управления средствами	

	аппаратного обеспечения информационных и автоматизированных систем	программной разработки	
ПК-5. Владение навыками использования различных технологий разработки программного обеспечения	ПК-5.1. Знает современные технологии разработки ПО (структурное, объектно-ориентированное)	Знает современные технологии разработки ПО (структурное, объектно - ориентированное) Умеет использовать современные технологии разработки ПО Имеет навыки использования современных технологий разработки ПО	Устный опрос, письменный опрос
	ПК-5.2. Умеет использовать современные технологии разработки ПО		
	ПК-5.3. Имеет навыки использования современных технологий разработки ПО		
ПК-6. Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества	ПК-6.1. Знает концепции и атрибуты качества ПО	Знает концепции и атрибуты качества ПО Умеет определять атрибуты качества ПО Имеет навыки в использовании методов, инструментов и технологий обеспечения качества ПО	Устный опрос, письменный опрос
	ПК-6.2. Умеет определять атрибуты качества ПО		
	ПК-6.3. Имеет навыки в использовании методов, инструментов и технологий обеспечения качества ПО		

4. Объем, структура и содержание дисциплины.

4.1. Объем дисциплины составляет 8 зачетных единиц, 288 академических часов.

4.2. Структура дисциплины.

4.2.1. Структура дисциплины в очной форме

№ п/п	Разделы и темы дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Самостоятельная работа	Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				Лекции	Практические занятия	Лабораторные занятия	Контроль самост. раб.		
Модуль 1. Понятия программной инженерии									
1	Введение в технологию программирования	3	1	4		4		12	Устный опрос
2	Элементы программной инженерии	3	2	2		2		12	Устный опрос
	<i>Итого по модулю 1:</i>			6		6		24	
Модуль 2. Визуальное моделирование на основе UML Методология (MSF)									
3	Визуальное моделирование при анализе и проектировании. Основы UML	3	3-4	2		2		6	Устный опрос
4	Методология MSF. Версии. Модель проектной группы	3	5	4		4		4	
5	Методология MSF. Управление рисками. Модель процессов	3	6	4		4		6	
	<i>Итого по модулю 2:</i>			10		10		16	
Модуль 3. Разработка и оценка качества программного обеспечения									
6	Методология MSF. Выработка концепции. Планирование	4	7	2		2		10	Устный опрос
7	Методология MSF. Фазы разработки и стабилизации	4	8-9	4		4		10	
	<i>Итого по модулю 3:</i>			8		8		20	
Модуль 4. Групповая разработка, управление версиями.									

8	Параллельная и конкурентная разработка.	4		6		6		8	Устный опрос
9	Различные способы организации коллектива разработчиков.	4		4		4		8	
	<i>Итого по модулю 4:</i>			10		10		16	
	<i>Всего часов за 3 семестр</i>			34		34		76	
<i>Модуль 5. Компонентные технологии и разработка распределённого ПО</i>									
10	Программные компоненты и интерфейсы	4		4		4		8	Письменный опрос
11	Компонентная модель разработки ПО	4		6		6		8	
	<i>Итого по модулю 5:</i>			10		10		16	
<i>Модуль 6. Тестирование, обеспечение качества</i>									
12	Виды тестирования	4		2		2		4	Устный опрос
13	Основы ручного тестирования	4		2		2		4	
14	Основы тест-аналитики	4		4		2		4	
15	Модульное тестирование	4		4		2		4	
	<i>Итого по модулю 6:</i>			12		8		16	
<i>Модуль 7: Сопровождение ПО</i>									
16	Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.	4		6		8		2	Письменный опрос
17	Реинжиниринг программных систем	4		2		8		2	
	<i>Итого модуль 7:</i>			12		16		8	
<i>Модуль 8. Подготовка к экзамену</i>								36	
	<i>Всего часов за 4 семестр:</i>			34		34		76	

4.3. Содержание дисциплины, структурированное по темам (разделам).

4.3.1. Содержание лекционных занятий по дисциплине

Модуль 1. Понятия программной инженерии

Тема 1. Введение в технологию программирования

- Основные понятия. Программирование. IT-проекты. Программы и программное обеспечение (программные продукты)
- Бизнес и IT-проекты. Рынок ПО в России и в мире.
- Причины неудачи IT-проектов
- Технологии программирования: структурное программирование, модульное программирование, объектно-ориентированное программирование, компонентное программирование

Тема 2. Элементы программной инженерии

- Программная инженерия, основные понятия. Инженеры и программные инженеры. Программная инженерия как инженерная дисциплина. Область действия программной инженерии. Цели программных инженеров. Программные инженеры и научная среда
- Процесс создания программного обеспечения. Понятие процесса. Модели процесса

Модуль 2. Визуальное моделирование на основе UML. Методология MSF

Тема 3. Визуальное моделирование при анализе и проектировании. Основы

Unified modeling language (UML)

- Анализ и проектирование. Некоторые частные вопросы. Обзор принципов объектного подхода. Повторное использование
- Визуальное моделирование. История языка UML. Вместо введения. Идея визуального моделирования. История языка UML
- Структура языка UML. Модели UML. Диаграммы UML. Понятия UML
- Учебный пример. Постановка задачи. Система бронирования билетов для авиакомпании
- Визуальное описание функциональной модели средствами UML. Актеры и варианты использования в UML
- Структура системы и ее описание средствами UML. Классы. Шаблоны классов. Объекты. Интерфейсы. Пакеты. Подсистемы. Компоненты. Комментарии. Отношения между элементами модели

Тема 4. Методология Microsoft solutions framework. Версии. Модель проектной группы

Введение в методологию MSF. Основные концепции методологии MSF 4.0. Направления в MSF 4.0. Основные положения MSF for Agile Software Development. Инструментальная поддержка MSF 4.0

Формирование команды. Модель проектной группы MSF for Agile Software Development. Основные принципы построения команды. Ролевые группы и роли, зоны ответственности ролевых групп. Рекомендации по возможному объединению ролей. Учебный пример. Формирование команды

Тема 5. Методология MSF. Управление рисками. Модель процессов

- Вспоминая предыдущую лекцию
- Управление рисками в MSF for Agile Software Development. Основные сведения о рисках. Планирование управления рисками. Процесс управления рисками. Управление рисками как составная часть жизненного цикла проекта. Учебный пример. Выделение рисков
- Модель процессов MSF for Agile Software Development. Принципы модели процессов.

Управление компромиссами. Схема процесса разработки

Модуль 3. Разработка и оценка качества программного обеспечения

Тема 6. Методология MSF. Выработка концепции. Планирование

- Вспоминая предыдущую лекцию
- Старт проекта. Фаза выработки концепции
- Планирование проекта. Фаза планирования

Тема 7. Методология MSF. Фазы разработки и стабилизации

- Вспоминая предыдущую лекцию
- Разработка решения. Фаза разработки
- Стабилизация решения. Фаза стабилизации
- Внедрение решения. Фаза внедрения

Модуль 4. Групповая разработка, управление версиями.

Тема 8. Параллельная и конкурентная разработка

Единый репозиторий проекта. Системы RCS, CVS. Методы нумерации версий. Параллельная и конкурентная разработка.

Тема 9. Различные способы организации коллектива разработчиков.

Организация коллектива разработчиков: матричный метод, метод главного специалиста, вертикальные и горизонтальные координации управления проектом.

Основные и вспомогательные подразделения и их задачи.

Модуль 5. Компонентные технологии и разработка распределённого ПО

Тема 10. Программные компоненты и интерфейсы

Программный компонент. Программный интерфейс. Программный контракт, предусловия и постусловия.

Тема 11. Компонентная модель разработки ПО

Компонентная модель, компонентная среда, базовые службы компонентной среды, распределённое ПО, прозрачность, открытость, масштабируемость, безопасность, синхронное и асинхронное взаимодействие, удалённый вызов процедур, транзакция.

Модуль 6. Тестирование, обеспечение качества

Тема 12. Виды тестирования

Модульные тесты. Интеграционное тестирование. Функциональное тестирование. Нефункциональные виды тестирования. Связанные с изменениями виды тестирования

Тема 13. Основы ручного тестирования

Цели, типы ручного тестирования. Инструменты для автоматизации ручного тестирования

Тема 14. Основы тест-аналитики

Тестовая комбинаторика. Тестирование состояний и переходов. Продвинутое тестирование состояний и переходов

Модуль 7. Сопровождение ПО

Тема 15. Исправление ошибок, внесение дополнительной функциональности, повышение эффективности.

Требования, предъявляемые к ПО и документации для реализации успешного сопровождения.

Тема 16. Реинжиниринг программных систем

Перевод устаревших программ на новые языки и платформы, возвратное проектирование – извлечение знаний из текста программы.

4.3.2. Содержание лабораторных занятий по дисциплине.

Темы практических занятий

Тема 1. C++Builder 2010 и современные информационные технологии

План занятия

1. Объектно-ориентированное программирование C++ Builder 2010
2. Основы визуального программирования интерфейса
3. Взаимодействие приложений в информационных системах
4. Распределенные многозвенные приложения
5. Переносимость данных и программ
6. Сетевые службы

Тема 2. Объектно-ориентированное проектирование в IDE C++ Builder 2010

План занятия

1. Общие сведения о программах на C++ Builder 2010
2. Структура головного файла проекта
3. Структура файлов модулей форм
4. Области видимости и доступ к объектам модуля
5. Указатели на объекты
6. Общий вид окна IDE
7. Главное меню. Быстрые кнопки
8. Палитра компонентов
9. Окно формы
10. Окно Редактора Кода
11. Инспектор Объектов
12. Перетаскивание и встраивание окон в IDE C++Builder 2010

Тема 3. Компоненты ввода и отображения информации

План занятия

1. Компоненты Label, StaticText, Panel
2. Окна редактирования Edit, LabeledEdit и MaskEdit
3. Многострочные окна редактирования Memo и RichEdit
4. Компоненты выбора из списков - ListBox, CheckListBox, ValueListEditor, ComboBox,
5. Таблица строк - компонент StringGrid
6. Компоненты ввода и отображения целых чисел - Up Down и CSpinEdit
7. Компоненты ввода и отображения дат и времени - DateTimePicker, MonthCalendar, SCalendar
8. Компонент генерации страницы Excel - FIBook
9. Компонент отображения иерархических данных - ListView

Тема 4. Кнопки, индикаторы, управляющие элементы

План занятия

1. Общая характеристика управляющих элементов
1. Управляющие кнопки Button и BitBtn
2. Кнопка с фиксацией SpeedButton
3. Группы радиокнопок - RadioGroup, RadioButton и GroupBox
4. Индикаторы CheckBox и CheckListBox
5. Ползунки и полосы прокрутки - компоненты TrackBar и ScrollBar
6. Таймер - компонент Timer
7. Заголовки - компоненты HeaderControl и Header

Тема 5. Системные диалоги

План занятия

1. Общая характеристика компонентов – диалогов
2. Диалоги открытия и сохранения файлов - компоненты OpenFileDialog и SaveDialog
3. Диалог выбора шрифта - компонент FontDialog
4. Диалоги поиска и замены текста -компоненты FindDialog и ReplaceDialog
5. Диалоги выбора цвета -компоненты ColorDialog
6. Диалоги печати и установки принтера -компоненты PrintDialog и PrinterSetupDialog

Тема 6. Проектирование графических, мультимедиа и анимационных приложений

План занятия

1. Построение графических изображений - компонент Image
2. Канва - холст для рисования
3. Режимы рисования. События OnPaint
4. Процедуры воспроизведения звуков Windows
5. Начала анимации - создание мультимедиа
6. Универсальный проигрыватель MediaPlayer
7. Воспроизведение немых видео клипов - компонент Animate

Тема 7. Архитектура приложений для локальных баз данных в C++ Builder 2010

План занятия

1. Модели баз данных
2. Организация связи с базами данных в C++Builder 2010
3. Обзор компонентов, используемых для связи с базами данных
4. Наборы данных Table.Основные свойства и события.
5. Компоненты визуализации и управления данными
6. Проектирование приложений с несколькими связанными таблицами
7. Состояние набора данных, пересылка записи в базу данных
8. Методы доступа к полям, навигации и поиска записей
9. Методы установки диапазона допустимых значений
10. Методы создания и модификации таблиц

Тема 8. Основы языка SQL, создание приложений для работы с базами данных в сети

План занятия

1. Оператор выбора Select
2. Операции с записями
3. Операции с таблицами
4. Операции с индексами
5. Основные свойства компонента Query
6. Основные методы и события компонента Query
7. Работа с базами данных в сети
8. InterBase - работа на платформе клиент/сервер
9. Доступ к базам данных через ADO
10. Обзор компонентов наборов данных
11. Доступ к InterBase через InterBase Express
12. Доступ к базам данных с помощью компонентов dbExpress
13. Технология MIDAS

Тема 9. Проектирование приложений для Интернет

План занятия

1. Создание собственного браузера
2. Динамические страницы Web - приложения CGI

3. Сервер Web C++ Builder 2010
4. Использование форм и таблиц в HTML
5. Использование шаблонов HTML
6. Использование активных форм
7. Обзор дополнительных возможностей работы с Интернет

Лабораторная работа № 1. Общие замечания

Процесс создания программы в C++Builder состоит из двух шагов: сначала нужно создать форму программы (диалоговое окно), а затем функции обработки *событий*. Форма приложения Windows создается путем добавления в нее *компонентов* и последующей их настройки.

В форме практически любого приложения есть компоненты, которые обеспечивают интерфейс между программой и пользователем. Такие компоненты называют базовыми. К базовым компонентам относятся:

- Label - поле вывода текста; Edit - поле редактирования текста;
- Button - командная кнопка; CheckBox - независимая кнопка выбора;
- RadioButton - зависимая кнопка выбора; ListBox - список выбора;
- ComboBox - комбинированный список выбора.

Вид компонента, его размер и поведение определяют значения *свойств* (характеристик) компонента.

Основную работу в программе выполняют функции обработки *событий*.

Исходную информацию программа может получить из полей редактирования (компонент Edit), списка выбора (компонент ListBox) или комбинированного списка (компонент ComboBox). Для ввода значений логического типа можно использовать Компоненты CheckBox и RadioButton.

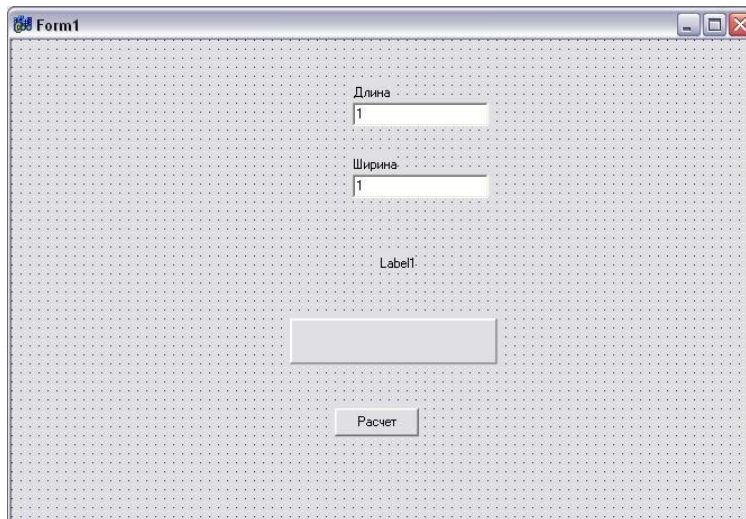
- Результат программа может вывести в поле вывода текста (компонент Label) или в окно сообщения (функции ShowMessage, MessageDlg).
- Для преобразования текста, например, находящегося в поле редактирования, в целое число нужно использовать функцию StrToInt, а в дробное - функцию StrToFloat. Для преобразования целого, например, значения переменной, в строку нужно использовать функцию IntToStr, а для преобразования дробного - функцию FloatToStr или FloatToStrF.

Создадим приложение, которое при нажатии кнопки перемножает два числа, введенных пользователем, и показывает результат умножения. Эти числа будем понимать, как длину и ширину сторон прямоугольника, и тогда результат - это площадь.

При построении этого приложения используйте компоненты - окна редактирования **LabeledEdit**. Результат нужно выводить не в метку **Label**, а в панель **Panel**, чтобы испытать новый компонент.

1. Откройте новое приложение.

2. Перенесите на форму со страницы библиотеки **Additional** два окна компонента **LabeledEdit**, а со страницы библиотеки **Standard** - одну панель **Panel**, одну кнопку **Button** и одну метку **Label** для надписи. Разместите все это примерно так, как показано на рисунке.



3. Измените надписи в метках компонентов **LabeledEdit** на «Ширина», «Высота». Для этого щелкните на символе "+" в свойстве **EditLabel** этих компонентов и измените надпись в свойстве **Caption** раскрывшихся списков свойств меток. Задайте для меток жирный шрифт.

4. Измените свойство **Caption** кнопки на «Расчет». Очистите свойство **Caption** у панели.

В свойстве **Text** компонентов **LabeledEdit** задайте 1" - начальное значение текста. Установите свойства **BevelInner = bvLowered** и **BevelOuter = bvRaised** панели, которые определяют вид (утопленный - **bvLowered** или выпуклый **bvRaised**) основного поля и рамки панели.

Напишите обработчик щелчка кнопки. Операторы этого обработчика имеют вид:

`Label1->Caption="Площадь прямоугольника равна:";`

`Panel1->Caption = LabeledEdit1->Text + " * " + LabeledEdit2->Text + " = " + FloatToStr(StrToFloat(LabeledEdit1->Text) * StrToFloat(LabeledEdit2->Text));`

Во втором операторе свойству **Caption** компонента **Panel1** присваивается значение выражения, указанного в правой части оператора. Это выражение должно иметь тип строки текста. Начинается строка с текста, введенного пользователем в окно редактирования **Edit1** - этот текст хранится в свойстве **Text**. Затем прибавляете к этому тексту символы " * ". Знак "+" в выражениях для строк означает конкатенацию - сцепление двух строк символов. Затем аналогичным образом к строке добавляется текст второго окна редактирования и символы " = ". После вставляется результат перемножения двух целых чисел. Этот результат будет числом и, чтобы вставить его в текст, надо сначала преобразовать это число в строку. Эту операцию выполняет функция **FloatToStr(...)**, которая преобразует в строку само произведение двух чисел. Но числа заданы пользователем в виде текстов - строк символов в окнах редактирования. Прежде, чем перемножать, эти строки надо перевести в числа. Эту операцию выполняют функции **StrToFloat()**, преобразующие символьное изображение числа в его значение типа действительного числа. Знак '*', указанный между двумя функциями **StrToFloat**, обозначает операцию умножения.

Разработка проекта «Любимые напитки»

Цель работы

Разработать проект с использованием объектов **Listbox**.

Спецификация программы:

На форме должны быть расположены 3 списка. Один из списков содержит названия напитков. Пользователь может переносить элементы из этого списка в списки «Любимые» и «Нелюбимые» и обратно. При этом перемещаемый элемент должен удаляться из списка-источника. То есть, например, перенос элемента «Чай» из общего списка в список «Любимые» происходит в следующем порядке:

он добавляется в список «Любимые» и удаляется из общего списка. Таким образом общее количество элементов всех трех списков остается постоянным. Перенос элементов между списками должен осуществляться по нажатию на соответствующие кнопки, либо мышью (система Drag and Drop).

Система Drag and Drop позволяет напрямую перетаскивать объекты между разными источниками, например, из одного списка в другой. «Перетаскивание» представляет собой нажатие и удерживание левой кнопки мыши на объекте и дальнейшее его перемещение за курсором в желаемую область.

Списки «Любимые» и «Нелюбимые» должны сохраняться в текстовые файлы.

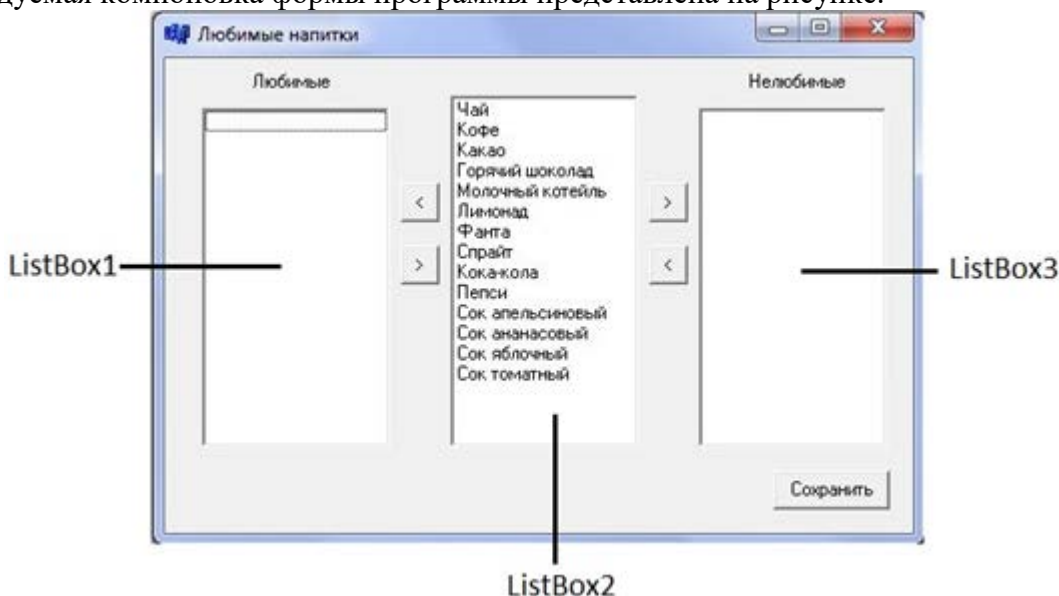
Для создания формы использовать компоненты:

Label – для подписей

ListBox – для вывода списков

Button – для инициирования действий

Рекомендуемая компоновка формы программы представлена на рисунке.



Рекомендуемая форма проекта

5. Образовательные технологии

Основными образовательными технологиями проведения курса «Технологии программирования» являются:

- Лекции, сопровождаемые компьютерными презентациями;
- лабораторные работы, в рамках которых составляются и тестируются программы, иллюстрирующие теоретический материал лекций;
- самостоятельная работа студентов, включающая усвоение теоретического материала, поиск дополнительного материала и эффективных способов выполнения заданий, завершение выполнения лабораторных работ; оформление и подготовка к защите лабораторных работ, подготовка к текущему контролю знаний и к итоговому экзамену;
- разработанные индивидуальные задания для самостоятельной работы;
- рейтинговая технология контроля учебной деятельности студентов для обеспечения их ритмичной работы в течение семестра
- консультирование студентов по вопросам учебного материала и выполнения курсового заданий.

6. Учебно-методическое обеспечение самостоятельной работы студентов

В процессе обучения студентов применяются следующие формы контроля успеваемости:

- посещаемость лекций;
- посещаемость лабораторных занятий;
- выполнение и сдача лабораторных заданий.

Примерное распределение времени самостоятельной работы студентов

Вид самостоятельной работы	Примерная трудоёмкость	Формируемые компетенции
Текущая СРС		
Подготовка к лекции, работа с учебной литературой и электронными источниками	10	ОПК-5
Подготовка к практическим, лабораторным занятиям	10	ПК-5, ПК-6
подготовка к контрольным работам	20	ОПК-5, ПК-5, ПК-6
выполнение домашних заданий в виде решения отдельных задач, расчетно - компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин	16	ОПК-5, ПК-5, ПК-6
самостоятельное изучение разделов дисциплины	10	ПК-5, ПК-6
Творческая проблемно-ориентированная самостоятельная работа		
поиск, изучение и презентация информации по заданной проблеме	10	ОПК-5, ПК-5, ПК-6
Подготовка письменных работ (рефератов)	20	ПК-5
исследовательская работа, участие в конференциях, семинарах, олимпиадах	20	ОПК-5, ПК-5, ПК-6
Итого СРС	116	

7. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

7.1. Типовые контрольные задания

1. Характеристика отрасли производства программного обеспечения
2. Технологический цикл разработки программных систем (ЖЦ ПО)
3. Стадии и этапы разработки программного обеспечения
4. Анализ требований, предъявляемых к системе
5. Разработка технического задания
6. Разработка проекта (развернутый план проекта)
7. Классический жизненный цикл
8. Макетирование
9. Стратегии конструирования ПО
10. Инкрементная модель
11. Быстрая разработка приложений

12. Спиральная модель
13. Компонентно–ориентированная модель
14. XP – процесс
15. Структура идеального XP - процесса
16. Стандарты документирования программных средств. Виды программных документов.
17. Единая система программной документации
18. Методология проектирования программных систем
19. Информационное моделирование
20. Проектирование интерфейса пользователя (Диалоговый интерфейс)
21. Проектирование интерфейса пользователя (Графический интерфейс)
22. Стил программирования. Модульность, структурность.
23. Стил программирования. Структурированность, читабельность.
24. Структура программных продуктов
25. Модульное программирование
26. Нисходящее проектирование
27. Структурное программирование. Принципы структурной методологии
28. Связность модуля
29. Сцепление модулей
30. Основные понятия объектно–ориентированного программирования
31. Методика объектно-ориентированного программирования
32. Этапы программирования в ООП
33. Принципы программирования в ООП
34. Критерии эффективного программирования
35. Оптимизация программ
36. Отладка программного обеспечения
37. Рекомендации по организации отладки
38. Отладочные средства C++ и C#
39. Тестирование, правила, этапы, виды
40. Ошибки ПО, заповеди тестирования, психологические аспекты тестирования
41. Тестирование программы как «чёрного ящика», как «белого ящика»
42. Тестирование модулей. Пошаговое тестирование
43. Восходящее тестирование, нисходящее тестирование

7.2. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.

Общий результат выводится как интегральная оценка, складывающаяся из текущего контроля - 50% и промежуточного контроля - 50%.

Текущий контроль по дисциплине включает:

- посещение занятий – 5 баллов,
- участие на практических занятиях - 20 баллов,
- выполнение лабораторных заданий – 60 баллов,
- выполнение домашних (аудиторных) контрольных работ –15 баллов.

Промежуточный контроль по дисциплине включает:

- устный опрос - 30 баллов,
- письменная контрольная работа - 70 баллов.

Итоговой формой контроля знаний, умений и навыков по дисциплине является экзамен. Экзамен проводится в форме устного опроса. При соответствии ответа учащегося на экзамене более чем 51 % критериев из этого списка выставляется оценка «удовлетворительно», 66% – 85% оценка «хорошо», 86% и выше оценка «отлично».

Критерии оценки экзамена по 100-бальной системе:

- 100 баллов - дан полный, развёрнутый ответ на поставленный вопрос, показана совокупность осознанных знаний об объекте, проявляющаяся в свободном ориентировании понятиями, умении выделять существенные и несущественные его признаки, причинно-следственные связи. Ответ формулируется в терминах науки, логичен, доказателен, демонстрирует авторскую позицию студента.

- 90 баллов - дан полный, развёрнутый ответ на поставленный вопрос, показана совокупность осознанных знаний об объекте, доказательно раскрыты основные положения темы; в ответе прослеживается чёткая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Ответ изложен литературным языком в терминах науки. Могут быть допущены недочёты в определении понятий, исправленные студентом самостоятельно в процессе ответа.

- 80 баллов - дан полный, развёрнутый ответ на поставленный вопрос, доказательно раскрыты основные положения темы; в ответе прослеживается чёткая структура, логическая последовательность, отражающая сущность раскрываемых понятий, теорий, явлений. Ответ изложен литературным языком в терминах науки. Могут быть допущены недочёты, исправленные студентом с помощью преподавателя.

- 70 баллов - дан полный, но недостаточно последовательный ответ на поставленный вопрос, но при этом показано умение выделить существенные и несущественные признаки и причинно-следственные связи. Ответ логичен и изложен в терминах науки. Могут быть допущены 1-2 ошибки в определении основных понятий, которые студент затрудняется исправить самостоятельно.

- 60 баллов - дан неполный ответ, логика и последовательность изложения имеют существенные нарушения. Допущены грубые ошибки при определении сущности раскрываемых понятий, теорий, явлений, вследствие непонимания студентом их существенных и несущественных признаков и связей. В ответе отсутствуют выводы. Умение раскрыть конкретные проявления обобщённых знаний не показано. Речевое оформление требует поправок, коррекции.

- 50 баллов - дан неполный ответ, представляющий собой разрозненные знания по теме вопроса с существенными ошибками в определениях. Присутствует фрагментарность, нелогичность изложения. Не понимает связь данного понятия, теории, явления с другими объектами дисциплины. Отсутствуют выводы. Конкретизация и доказательность изложения. Речь неграмотная. Дополнительные и уточняющие вопросы преподавателя не приводят к коррекции ответа студента не только на поставленный вопрос, но и на другие вопросы дисциплины.

- 40 баллов - ответ студента правилен лишь частично, при разъяснении материала допускаются серьезные ошибки.

- 20-30 баллов - студент имеет общее представление о теме, но не умеет логически обосновать свои мысли.

- 10 баллов - студент имеет лишь частичное представление о теме.

- 0 баллов – нет ответа.

8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

а) адрес сайта курса - <http://edu.dgu.ru/course/view.php?id=3285>

б) основная литература:

1. Васильев, В. Н. Основы программирования на языке C#: учебное пособие / В. Н. Васильев. — Волгоград: Волгоградский институт бизнеса, 2010. — 70 с. — ISBN 978-5-9061-7234-1. — Текст: электронный // Электронно-библиотечная система IPR BOOKS [сайт]. — URL: <https://www.iprbookshop.ru/11341.html> (дата обращения: 29.11.2021).

2. Гагарина Л. Г. Технология разработки программного обеспечения: [учеб. пособие] / Гагарина, Лариса Геннадьевна, Е. В. Кокорева; под ред. Л.Г.Гагариной. - М. : ФОРУМ: ИНФРА-М, 2019 399 с. - (Высшее образование). - Допущено УМО. - ISBN 978-5-8199-0342-1 (ИД "ФОРУМ"): 246-84.

3. Ковалевская, Е. В. Методы программирования: учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — Москва: Евразийский открытый институт, 2011. — 320 с. — ISBN 978-5-374-00356-7. — Текст: электронный // Электронно-библиотечная система IPR BOOKS: [сайт]. — URL: <https://www.iprbookshop.ru/10784.html> (дата обращения: 29.11.2021).

в) дополнительная литература

1. Анашкина Н.В., Петухова Н.Н., Смольянинов В.Ю. Технологии и методы программирования

2. Г. Буч, Дж. Рамбо, А. Джекобсон. UML. Руководство пользователя. – ДМК-Пресс, Питер, 2004.

3. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: серия «Высшее образование», М.: Изд-во: «Форум, Инфра-М», 2019

4. Жоголев А.А. Технологии программирования. Компонентный подход. – М.: Научный мир, 2008

5. Иан Соммервиль. Инженерия программного обеспечения. 6 изд, и.д. "Вильямс", 2018. — 624 с.

6. Иванова Г. С Технология программирования: Учебник для вузов Изд. 3-е, перераб., доп. 3-е, стереотип. / Иванова Г. С. - М.: Изд-во МГТУ им. Н.Э. Баумана, 2008

7. Кулямин В.В. Технологии программирования. Компонентный подход. – СПб.: Питер, 2014 г.

8. 1846A: Microsoft Solutions Framework Essentials. Microsoft Official Course, 2009

9. 2710B: Analyzing Requirements and Defining Microsoft .NET Solutions Architecture. Microsoft Official Course, 2009

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. ЭБС IPRbooks: <http://www.iprbookshop.ru/>

2. Электронно-библиотечная система «Университетская библиотека онлайн»(архив):www.biblioclub.ru

3. Единое окно доступа к образовательным ресурсам. <http://window.edu.ru/>

4. <http://www.microsoft.com/msf>

5. <http://www.uml.org>

6. <http://www.wikipedia.org>

7. <http://www.wikipedia.org>

8. MSF for Agile Software Development Process Guidance: [\[http://go.microsoft.com/fwlink/?linkid=63524\]](http://go.microsoft.com/fwlink/?linkid=63524)

9. Алистер Кокбёрн. Каждому проекту своя методология:

[\[http://software-testing.ru/lib/cockburn/methodology-per-project.htm\]](http://software-testing.ru/lib/cockburn/methodology-per-project.htm)

[\[http://alistair.cockburn.us/index.php/Methodology_per_project\]](http://alistair.cockburn.us/index.php/Methodology_per_project)).

10. Методические указания для обучающихся по освоению дисциплины.

При освоении всех разделов дисциплины необходимо сочетание всех форм учебной деятельности: изучение лекционного материала, выполнение заданий на практических занятиях и лабораторных работах, как с использованием компьютера, так и без него, самостоятельная работа с рекомендуемой литературой и использование методических указаний, консультации преподавателя при выполнении расчетно-графических работ.

После каждого лекционного занятия студенты должны повторить материал лекции по конспектам, а перед каждым очередным занятием - освежить в памяти материал предыдущего.

Самостоятельная работа ориентирует студентов на углубленное изучение и осмысление тем учебного курса. При подготовке к лабораторной работе студент должен изучить рекомендуемые материалы. Если в задании на лабораторную работу есть непонятные неясные моменты, необходимо задать вопросы преподавателю. По каждой лабораторной работе необходимо подготовить отчет, в котором отразить все основные действия, выполняемые в процессе лабораторной работы, а также результаты, полученные при выполнении лабораторной работы

11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем.

В процессе преподавания дисциплины предполагается использование современных технологий визуализации учебной информации (создание и демонстрация презентаций), использование ресурсов электронной информационно-образовательной среды университета.

При проведении занятий по данной дисциплине используется программное обеспечение - Embarcadero Rad Studio и Visual Studio Community 2022.

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине.

Лекции и практические занятия по данной дисциплине проводятся в мультимедийном лекционном зале, где установлен проектор и экран. Практические занятия проводятся в двух компьютерных классах где установлены по 15 компьютеров, все они подключены локальной сети университета т.е. имеют доступ к локальным ресурсам ДГУ и глобальной сети Интернет.