

МИНОБРНАУКИ РОССИИ



Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(ФГБОУ ВО «РГГУ»)

ФАКУЛЬТЕТ «ИНФОРМАЦИОННЫХ СИСТЕМ И БЕЗОПАСНОСТИ»
Кафедра «Информационных технологий и систем»

**ПРОГРАММИРОВАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ
ИНФОРМАЦИОННЫХ СИСТЕМ ГУМАНИТАРНОЙ СФЕРЫ**

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Направление подготовки 09.03.03 «Прикладная информатика»
Направленность: Прикладная информатика в гуманитарной сфере

Уровень высшего образования: бакалавриат

Форма обучения: очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2023

Программирование пользовательских интерфейсов информационных систем
гуманитарной сферы

Рабочая программа дисциплины

Составитель:

к.х.н., с.н.с., доцент Подорожный А.М.

Ответственный редактор

к.с-х.н., доцент, зав. кафедрой

Информационных технологий и систем Н.Ш. Шукенбаева

УТВЕРЖДЕНО

Протокол заседания кафедры

Информационных технологий и систем

№ 8 от 15.04.2023 г.

ОГЛАВЛЕНИЕ

1. Пояснительная записка.....	4
1.1. Цель и задачи дисциплины	4
1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине.....	4
1.3. Место дисциплины в структуре образовательной программы	5
2. Структура дисциплины.....	5
3. Содержание дисциплины	5
4. Образовательные технологии	6
5. Оценка планируемых результатов обучения.....	7
5.1. Система оценивания	7
5.2. Критерии выставления оценки по дисциплине	7
5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	9
6. Учебно-методическое и информационное обеспечение дисциплины.....	9
6.1. Список источников и литературы.....	9
6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».	10
6.3. Профессиональные базы данных и информационно-справочные системы	11
7. Материально-техническое обеспечение дисциплины	11
8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья.....	11
9. Методические материалы.....	13
9.1. Планы практических работ	13
9.2. Задания для самостоятельной работы	18
Приложение 1. Аннотация дисциплины.....	21

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Цель дисциплины - приобретение знаний, навыков и умений в области программирования интерфейсов пользователя, овладение знаниями и навыками в программировании на языках высокого уровня, основами дизайна, эргономики и юзабилити пользовательских интерфейсов.

Задачи дисциплины:

- Изучение видов, средств и методов создания пользовательских интерфейсов;
- Изучение основ дизайна и эргономики интерфейсов;
- Практическая разработка GUI интерфейсов средствами C++ и Java.

1.2. Формируемые компетенции, соотнесённые с планируемыми результатами обучения по дисциплине

Коды компетенции	Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ПК-2 Способен разрабатывать, тестировать и адаптировать прикладное программное обеспечение	ПК-2.1. Знает основные среды для разработки программного обеспечения, методы тестирования и адаптации прикладного программного обеспечения.	Знать: технологии и методы программирования с использованием IDE в задачах разработки интерфейсов на языках Java и C++; методы принципы и средства проектирования графических интерфейсов пользователя (GUI), методы тестирования программных компонентов пользовательских интерфейсов на основе реализации юзабилити для потенциальных пользователей.
	ПК-2.2. Умеет разрабатывать, тестировать и адаптировать прикладное программное обеспечение.	Уметь: применять технологии и методы программирования с использованием IDE в задачах разработки интерфейсов GUI; использовать оптимальные принципы и методы проектирования интерфейсов в задачах информатизации гуманитарной сферы; тестировать компоненты программного обеспечения информационных систем на основе юзабилити человеко-компьютерного взаимодействия.
	ПК-2.3. Владеет современными языками программирования и методиками разработки, тестирования и адаптации прикладного программного обеспечения.	Владеть: технологиями и методами программирования с использованием IDE в задачах разработки интерфейсов на языках Java и C++; методами принципами и средствами проектирования графических интерфейсов пользователя (GUI) на основе достижений эргономики и дизайна; практическими навыками проектирования и тестирования пользовательских интерфейсов.

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Программирование пользовательских интерфейсов информационных систем гуманитарной сферы» относится к части дисциплин, формируемой участниками образовательных отношений учебного плана по направлению подготовки «Прикладная информатика».

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин: «Архитектура вычислительных систем», «Компьютерная графика в задачах информатизации гуманитарной сферы», «Объектно-ориентированное программирование», «Программирование Java / Python».

В результате освоения дисциплины формируются знания, умения и владения, необходимые для изучения следующих дисциплин и прохождения практик: «Проектирование информационных систем», «Web-дизайн в гуманитарной сфере», «Web-программирование в гуманитарной сфере».

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 3 з.е., 108 академических часов.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
4	Лекции	14
4	Практические занятия	28
Всего:		42

Объем дисциплины в форме самостоятельной работы обучающихся составляет 66 академических часов.

3. Содержание дисциплины

№	Наименование раздела дисциплины	Содержание
1	Пользовательские интерфейсы.	Интерфейс командной строки, готовые формы. Графический интерфейс (GUI), его значение, концепции WIMP, WYSIWYG, Drag-and-drop. Мультимедиа SILK интерфейс, реализация его элементов. Средства создания GUI интерфейсов: линейка Microsoft Visual Studio, средства Java, X Window System, FLTK, другие средства. Речевой интерфейс человек-компьютер, характеристики речи, идентификация человека по голосу, методы синтеза речи, конвертация текста в речь.
2	Последовательность разработки и	Модель эволюции прототипов, пошаговый план, UCD-дизайн, юзабилити. Уровни разработки интерфейса. Уровень стратегии: постановка

	структура пользовательских интерфейсов	задачи, технические требования, изучение аудитории, сегментация и профилирование, метод персонажей, пользовательские сценарии. Уровень набора возможностей, спецификация функциональности. Уровень структуры, интерфейсные шаблоны: прямого действия, сохранения контента, сокращения усилий, реализующие приглашения, шаблоны переходов. Уровень компоновки: интерфейсы программы и веб-сайта. Правила взаимодействия человек-компьютер: соответствие ментальной модели, понятность и однозначность, временные характеристики, средства осознанной работы с информацией. Документальное оформление интерфейса: техническое задание, роль технического писателя, ЕСПД, руководство пользователя, справочная система.
3	Дизайн и эргономика графических интерфейсов.	Дизайн, многозначность этого понятия, соотношение эстетической и функциональной сторон. Воздействие цветов на эмоции, круг Иттена, цвет в дизайне интерфейсов. Предпочтительные формы в создании интерфейсов, иллюзорность. Правила композиции, единство стиля, контраст, пропорции, соответствие задаче, проработка деталей. Дизайн текста, анимация, звуковой дизайн в пользовательских интерфейсах. Эргономика, области ее применения. Факторы совместимости человек-компьютер: сенсомоторные (время реакции человека, время усвоения информации, объем кратковременно памяти), антропометрические (физические условия работы, требования к безопасности, оптимальная поза работы, требования к монитору, клавиатуре, креслу, работа с мобильной техникой).
4	Программирование интерфейсов на C++ в API Windows Forms.	Начало работы с конструктором Windows Forms. Создание средств просмотра рисунков. Создание ограниченной по времени математической головоломки. Создание игры «Подбери пару». Создать калькулятор, аналогичный калькулятору Windows, вид – обычный. Создать приложение "Ежедневник". Возможности: создание записей двух видов: задача и заметка. Для задачи указывается срок выполнения. Заметка относится к одной из категорий. Список категорий можно изменять. Добавить поиск по содержанию заметок/задач в рамках выбранных категорий.
5	Программирование интерфейсов на Java в API NetBeans.	Введение в разработку графического интерфейса. Разработка Swing GUI в среде NetBeans. Обработка изображений в приложении с графическим интерфейсом. Интернационализация формы графического интерфейса. Бег. Пользователь указывает количество км за каждый день в течение некоторого периода времени. Программа находит общий пробег, среднее значение в день, наименьшее и наибольшее значения.

4. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5. Оценка планируемых результатов обучения

5.1. Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- отчет по практическим работам №1, 2,3, 4, 6, 8.	3	18
- отчет по практическим работам № 5, 7, 9	6	18
- опрос-коллоквиум		
Итого:	8	24
		60 баллов
Промежуточная аттестация: зачет с оценкой		40 баллов
Итого за семестр		100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55		E	
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2. Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	«отлично»/ «зачтено (отлично)»/ «зачтено»	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения. Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
82-68/ С	«хорошо»/ «зачтено (хорошо)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	«удовлетворительно»/ «зачтено (удовлетворительно)»/ «зачтено»	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>
49-0/ F,FX	«неудовлетворительно»/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами.</p> <p>Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3. Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

Вопросы к коллоквиуму №1

1. Пользовательские интерфейсы, командная строка, готовые формы. (ПК-2.1)
2. Графический интерфейс, SILK интерфейс. (ПК-2.1)
3. Создание графических интерфейсов средствами Microsoft и Java. (ПК-2.2)
4. Создание графических интерфейсов средствами X Window System и FLTK. (ПК-2.2)
5. Характеристики речи, идентификация человека по голосу. (ПК-2.2)
6. Методы синтеза речи. (ПК-2.3)
7. Подходы к разработке пользовательских интерфейсов. (ПК-2.3)
8. Уровень стратегии: постановка задачи, определение технических требований. (ПК-2.1)
9. Сегментация и профилирование пользовательской аудитории. (ПК-2.3)
10. Метод персонажей, пользовательские сценарии. (ПК-2.2)

Вопросы к коллоквиуму №2

1. Спецификация функциональности: назначение, содержание, средства представления, связь с ТЗ. (ПК-2.2)
2. Интерфейсные шаблоны: прямого действия, сохранения контента на странице. (ПК-2.2)
3. Интерфейсные шаблоны: сокращения усилий, приглашения. (ПК-2.2)
4. Интерфейсные шаблоны: переходы, мгновенная реакция интерфейса. (ПК-2.2)
5. Интерфейсы программных продуктов и веб-сайтов. (ПК-2.3)
6. Правила взаимодействия человек-компьютер: соответствие ментальной модели, понятность, однозначность. (ПК-2.3)
7. Правила взаимодействия человек-компьютер: временные характеристики, осознанной работы с информацией. (ПК-2.3)
8. Документальное оформление пользовательского интерфейса, (ПК-2.3) справочная система.

Вопросы к коллоквиуму №3

1. Определение, значения понятия: Дизайн. (ПК-2.3)
2. Эстетическая и функциональная стороны дизайна. (ПК-2.3)
3. Роль цвета в дизайне интерфейсов. (ПК-2.3)
4. Форма объектов в дизайне интерфейсов. (ПК-2.3)
5. Принципы композиции в дизайне интерфейсов. (ПК-2.1)
6. Текст, анимация, звук в дизайне интерфейсов. (ПК-2.1)
7. Эргономика, ее применение, сенсомоторные факторы в дизайне интерфейсов. (ПК-2.2)
8. Антропометрические факторы при работе на компьютере. (ПК-2.2)

На основе лекционного курса и практических заданий составлены тесты, свыше 100 вопросов. Тестовое задание для каждого студента состоит из 24 вопросов, отобранных по случайной выборке, пропорционально из каждого раздела. Тестовые задания разработаны для оценки остаточных знаний студентов.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1. Список источников и литературы

Основной

1. Магазанник, В. Д. Человеко-компьютерное взаимодействие : учебное пособие / В. Д. Магазанник. - 2-е изд., доп. - Москва : Университетская книга, 2020. - 408 с. - ISBN 978-5-98699-181-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1214481>
2. Дружинин, Д. В. Сжатие видеоданных графического интерфейса пользователя : учебное пособие / Д. В. Дружинин, А. В. Замятин. - Томск : Издательство Томского государственного университета, 2020. - 144 с. - ISBN 978-5-94621-934-1. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1864756>
3. А. В. Абрамян. М. Э. Абрамян. Разработка пользовательского интерфейса на основе системы Windows Presentation Foundation: учебник /; Южный федеральный университет. — Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. — 301 с. - ISBN 978-5-9275-2375-7. - Текст : электронный. - URL: <http://znanium.com/catalog/product/1020507>.
4. Нехорошкова, Л. Г. Информационное моделирование и анализ требований : учебное пособие / Л. Г. Нехорошкова. - Йошкар-Ола : ПГТУ, 2020. - 146 с. - ISBN 978-5-8158-2209-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1869363>
5. Ткаченко, О. Н. Взаимодействие пользователей с интерфейсами информационных систем для мобильных устройств: исследование опыта : учебное пособие / О.Н. Ткаченко. — Москва : Магистр : ИНФРА-М, 2022. — 152 с. - ISBN 978-5-9776-0457-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1859029>

Дополнительный

1. Информационные технологии и управляющие системы : монография / под науч. ред. док. техн. наук, проф. В.М. Артюшенко. - Москва : Научный консультант, 2015. — 184 с. - ISBN 978-5-9906953-8-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1023019>
2. Лиэнг, Ш. Интерфейс JNI: руководство по программированию / пер. с англ. И. Таловой. - Москва : ДМК Пресс, 2014. - 280 с. - ISBN 978-5-94074-982-0. - Текст : электронный. - URL: <http://znanium.com/catalog/product/1028105>
3. Impact Mapping. Как повысить эффективность программных продуктов и проектов по их разработке: Практическое руководство / Аджич Г. - М.:Альпина Паблишер, 2017. - 86 с.: 60x90 1/8 (Обложка) ISBN 978-5-9614-6448-1 - Режим доступа: <http://znanium.com/catalog/product/1003514>
4. Д. Гинсбург, Б. Пурномо. OpenGL ES 3.0. Руководство разработчика / пер. с англ. А. Борескова. - Москва: ДМК Пресс, 2015. - 448 с. - ISBN 978-5-97060-256-0. - Текст : электронный. - URL: <http://znanium.com/catalog/product/1027820>

6.2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет».

1. Vs.ru – самый популярный по числу посещений IT-ресурс. Публикуются новости, обзоры, аналитические вклады, модели заработка и пр.
2. С.news – крупнейшее издание в сфере высоких технологий на территории РФ и стран СНГ

3. <https://www.jstor.org> – полнотекстовая база данных рецензируемых научных журналов и книг, 1200 издателей из 60 стран мира
4. <http://www.consultant.ru/> - КонсультантПлюс компьютерная справочная правовая система.
5. <https://www.scopus.com> – Международная реферативная наукометрическая БД, Scopus, доступна по подписке.

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Информационные справочные системы:

1. Консультант Плюс
2. Гарант

7. Материально-техническое обеспечение дисциплины

Для реализации программы дисциплины требуется:

- компьютеры, с предустановленным базовым программным обеспечением в составе ОС Windows 10 и MS Office;
- объединение компьютеров в локальную сеть компьютерного класса с высокоскоростным выходом в интернет;
- для лекционного курса – посадочные места по количеству обучающихся, доска, мультимедиа проектор с экраном;
- рабочее место преподавателя, оснащенное компьютером с выходом на проектор.

Используемое программное обеспечение:

1. Windows 10
2. Microsoft Office 2013 Pro
3. Mozilla Firefox 52.8.1 ESR
4. Kaspersky Endpoint Security
5. Платформа ZOOM
6. Среда разработки Microsoft Visual Studio
7. Среда разработки NetBeans IDE 8.2
8. Виртуальная машина JRE 8

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере со специализированным программным обеспечением, или могут быть заменены устным ответом;
 - обеспечивается индивидуальное равномерное освещение не менее 300 люкс;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;

- письменные задания оформляются увеличенным шрифтом;
- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования;

- письменные задания выполняются на компьютере в письменной форме;
- экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;

- письменные задания выполняются на компьютере со специализированным программным обеспечением;

- экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
 - в форме аудиофайла.

- для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.

- для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа;
 - в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих:
 - устройством для сканирования и чтения с камерой SARA CE;
 - дисплеем Брайля PAC Mate 20;
 - принтером Брайля EmBraille ViewPlus;
- для глухих и слабослышащих:
 - автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих;
 - акустический усилитель и колонки;
- для обучающихся с нарушениями опорно-двигательного аппарата:

- передвижными, регулируемые эргономическими партами СИ-1;
- компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1. Планы практических работ

Практическая работа 1. Начало работы с конструктором Windows Forms.

1. Создание проекта пользовательского элемента управления.
2. Разработка макета элемента управления.
3. Заполнение элемента управления.
4. Использование окна структуры документа.
5. Добавление обработчиков событий.
6. Тестирование элемента управления.
7. Использование элемента управления в форме.

Практическая работа 2. Создание средства просмотра рисунков.

1. Создание проекта приложения Windows Forms.
2. Запуск приложения для просмотра изображений.
3. Настройка свойств формы.
4. Создание макета формы с помощью элемента управления TableLayoutPanel.
5. Добавление элементов управления в форму.
6. Присвоение имен элементам управления "Кнопка".
7. Добавление компонентов диалогового окна в форму.
8. Написание кода для обработчика событий кнопки "Показать рисунок".
9. Проверка, комментирование и тестирование кода.
10. Написание кода для дополнительных кнопок и флажка
11. Запуск приложения для просмотра изображений и изучение других функций

Практическая работа 3. Введение в разработку графического интерфейса..

1. Начало работы.
2. Знакомство со средством GUI Builder.
3. Ключевые понятия.
4. Первые действия.
5. Дальнейшие действия.
6. Обзор изученного.
7. Окончательная настройка.
8. Предпросмотр графического интерфейса.
9. Развёртывание приложения с графическим интерфейсом.

Практическая работа 4. Разработка Swing GUI в IDE NetBeans.

1. Создание проекта.
2. Создание внешнего интерфейса.
3. Добавление функциональности.
4. Выполнение программы.
5. Механизм обработки событий.

Практическая работа 5. Обработка изображений в приложении с графическим интерфейсом.

1. Введение.
2. Создание формы приложения.

3. Добавление пакета для изображения.
4. Вывод изображения на метке.
5. Сборка и запуск приложения.
6. Показ изображения как фона в кадре.
7. Создание пользовательского кода.

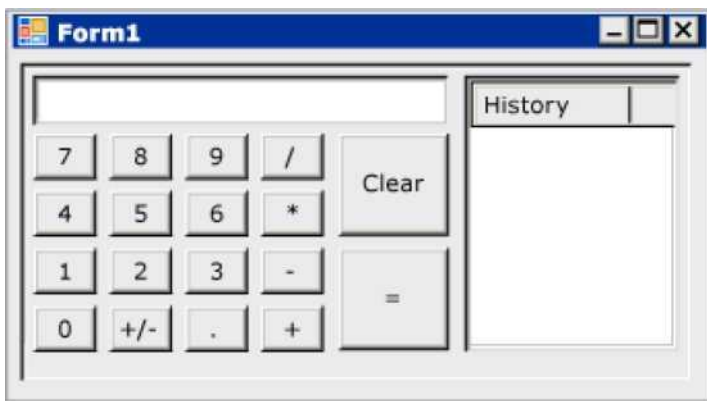
Практическая работа 6. Интернационализация формы графического интерфейса.

1. Принцип интернационализации.
2. Как настроить автоматическую интернационализацию на этапе проектирования.
3. Интернационализация всего проекта.
4. Интернационализация одной формы.
5. Дополнительные сведения.

Пример пошаговой инструкции к практической работе.

Работа 1.

Создание проекта пользовательского элемента управления



Конструктор Windows Forms предоставляет множество средств для создания приложений Windows Forms. Здесь создается пользовательский интерфейс для простого калькулятора. На рисунке показан макет элемента управления калькулятора.

1. Создание проекта «DemoCalculator»

- 1.1. Откройте **Visual Studio** и создайте проект категории **Библиотека элементов управления Windows Forms**.
- 1.2. Далее задайте проекту имя **DemoCalculatorLib** и папку, где он будет храниться.
- 1.3. Нажмите **Создать**, после чего появится интерфейс Visual Studio. Здесь на панели **Обозреватель решений** правой кнопкой мыши щелкните по **UserControl1.cs** и переименуйте его на **DemoCalculator1.cs**, а также все ссылки на элемент кода.

2. Разработка макета элемента управления

На этом этапе вы зададите расположение элементов управления.

- 2.1. Для окна на панели **Свойства** → **Size** задайте ширину **600** и высоту **400**.
- 2.2. Потяните за вкладку **Панель элементов**, открепив ее от верхней панели, выберите элемент управления **Контейнеры** → **SplitContainer** и перенесите **SplitContainer** в окно на **DemoCalculator1.cs**. В результате окно разделилось на две части.

Размер элемента управления **SplitContainer** автоматически устанавливается в соответствии с размером элемента управления **DemoCalculator**. В окне **Свойства** для свойства **Dock** задано значение **Fill**. Оно и устанавливает равенство размеров этих элементов.

- 2.3. Замените значение свойства **Dock** на **None**, теперь больше не будет меняться при изменении размера **DemoCalculator**. Это можно проверить, двигая **SplitContainer** за квадратики изменения его размеров.

- 2.4. Щелкните глиф смарт-тега (►) в правом верхнем углу **SplitContainer**. Выберите **Закрепить в родительском контейнере**, размеры элемента снова стали равными родительскому контейнеру, а **Dock** приобрело значение **Fill**.
- 2.5. Выберите вертикальную границу между панелями и перетащите ее вправо, так чтобы большую часть заняла левая из панелей (примерно, как на рисунке).
- SplitContainer разделяет DemoCalculator на две панели, границу между ними можно перемещать. На левой панели будут находиться кнопки калькулятора и экран, а на правой будет отображаться запись арифметических операций, выполненных пользователем.
- 2.6. В окне **Свойства** замените значение свойства **BorderStyle** на **Fixed3D**.
- 2.7. На панели элементов выберите элемент **Стандартные элементы управления** → **ListView** и перетащите его на левую панель элемента управления **SplitContainer1**.
- 2.8. Выберите глиф смарт-тега, замените значение параметра **Представление** на **Details**.
- 2.9. На панели смарт-тегов выберите **Изменить столбцы**, откроется диалоговое окно **Редактор коллекции columnHeader**. Здесь выберите **Добавить**, после чего в элемент **Text** впишите **Журнал**. Нажмите **ОК**, столбец создан.
- 2.10. На панели смарт-тегов выберите **Закрепить в родительском контейнере**, закройте панель смарт-тегов.
- 2.11. Аналогично перетащите элемент управления **TableLayoutPanel** на левую панель элемента **SplitContainer**.
- TableLayoutPanel упорядочивает свои дочерние элементы управления в сетке. Так будут созданы экран и кнопки элемента управления DemoCalculator.
- 2.12. На панели смарт-тегов выберите **Правка строк и столбцов**. Нажимайте кнопку **Добавить**, пока не добавятся пять столбцов. Выделите все пять столбцов, а затем в поле **Тип размера** выберите **Процент** и задайте значение **20**. При этом каждому столбцу задается одинаковая ширина.
- 2.13. В разделе **Показать** выберите **Строки**. Создайте пять строк, каждой строке задайте одинаковую высоту, **20%**. Закройте панель смарт-тегов.
- 2.14. В окне **Свойства** замените значение свойства **Dock** на **Fill**.

3. Заполнение элемента управления

Теперь, когда макет элемента управления настроен, можно добавить в элемент управления DemoCalculator кнопки и экран.

- 3.1. Дважды щелкните значок элемента управления **TextBox** на панели элементов. В результате он помещается в первую ячейку элемента **TableLayoutPanel**.
- 3.2. В окне **Свойства** замените значение **ColumnSpan** на **5**. Замените значение свойства **Anchor** на **Left, Right**. **TextBox** расширится по горизонтали, заняв все пять столбцов. Измените значение свойства **TextBox** на **Right**.
- 3.3. Разверните узел свойства **Font**, задайте значение **14** для **Size** и **true** для **Bold**.
- 3.4. Дважды щелкните значок **Button** на панели элементов, элемент **Button** разместится в следующей свободной ячейке. Далее дважды щелкните значок **Button** еще четыре раза, чтобы заполнить вторую строку элемента управления **TableLayoutPanel**.
- 3.5. Выберите все пять элементов управления **Button**, удерживая нажатой клавишу **SHIFT**. Скопируйте их в буфер обмена. Трехжды нажмите **CTRL+V**, получилось 20 кнопок в 20 ячейках.

- 3.6. Выделите все 20 элементов **Button**, после чего замените значение свойства **Dock** на **Fill**. Все кнопки полностью займут площадь ячеек.
- 3.7. Разверните узел свойства **Margin**, для **All** задайте значение **5**. Между кнопками увеличился размер полей.
- 3.8. Выберите **button1 0** и **button20**, клавишей **DELETE** удалите их. Выберите **button5** и **button15**, замените значение их свойства **Rowspan** на **2**. Это будут кнопки очистки и выдачи результата.

4. Использование окна структуры документа

Если в элементе управления или форме присутствует несколько элементов управления, перемещаться по макету удобнее с помощью окна "Структура документа".

4.1. В строке меню выберите **Вид → Другие окна → Структура документа**.

В этом окне структура **DemoCalculator** представлена в виде дерева. Окно позволяет переименовывать размещенные элементы управления.

4.2. В окне **Структура документа** щелкните правой кнопкой мыши элемент **button1** и замените его имя на **sevenButton**.

4.2. Замените имена элементов управления **Button**, заданные конструктором, на имена для рабочей среды согласно следующему списку:

- **button2** на **eightButton**;
- **button3** на **nineButton**;
- **button4** на **divisionButton**;
- **button5** на **clearButton**;
- **button6** на **fourButton**;
- **button7** на **fiveButton**;
- **button8** на **sixButton**;
- **button9** на **multiplicationButton**;
- **button11** на **oneButton**;
- **button12** на **twoButton**;
- **button13** на **threeButton**;
- **button14** на **subtractionButton**;
- **button15** на **equalsButton**;
- **button16** на **zeroButton**;
- **button17** на **changeSignButton**;
- **button18** на **decimalButton**;
- **button19** на **additionButton**;

4.3. С помощью окон **Структура документа** и **Свойства** измените свойство **Text** согласно списку:

- для элемента управления **sevenButton** замените свойство текста на **7**;
- для элемента управления **eightButton** замените свойство текста на **8**;
- для элемента управления **nineButton** замените свойство текста на **9**;
- для элемента управления **divisionButton** замените свойство текста на **/** (косая черта);
- для элемента управления **clearButton** замените свойство текста на **Clear**;

- для элемента управления **fourButton** замените свойство текста на **4**;
- для элемента управления **fiveButton** замените свойство текста на **5**;
- для элемента управления **sixButton** замените свойство текста на **6**;
- для элемента управления **multiplicationButton** замените свойство текста на * (звездочка);
- для элемента управления **oneButton** замените свойство текста на **1**;
- для элемента управления **twoButton** замените свойство текста на **2**;
- для элемента управления **threeButton** замените свойство текста на **3**;
- для элемента управления **subtractionButton** замените свойство текста на - (дефис);
- для элемента управления **equalsButton** замените свойство текста на = (знак равенства);
- для элемента управления **zeroButton** замените свойство текста на **0**;
- для элемента управления **changeSignButton** замените свойство текста на +/- ;
- для элемента управления **decimalButton** замените свойство текста на . (точка);
- для элемента управления **additionButton** замените свойство текста на + (знак "плюс");

4.4. На поверхности конструктора выберите все элементы управления **Button**, удерживая нажатой клавишу **SHIFT**.

4.5. В окне **Свойства** разверните узел **Font**. Задайте **14** для **Size** и **true** для **Bold** у всех выделенных элементов.

На этом разработка элемента управления DemoCalculator завершена. Остается только добавить логику калькулятора.

5. Добавление обработчиков событий

Кнопки в элементе управления DemoCalculator имеют обработчики событий, которые можно использовать для реализации большей части логики калькулятора. Конструктор Windows Forms позволяет реализовать заглушки всех обработчиков событий для всех кнопок одним двойным щелчком.

5.1. На поверхности конструктора выберите все элементы управления **Button**, удерживая нажатой клавишу **SHIFT**.

5.2. Откройте редактор кода (если его нет, то загрузите его через вкладку **Вид**).

5.3. Дважды щелкните один из элементов управления **Button**. Редактор кода меняется: в нем открылись обработчики событий, созданные конструктором.

6. Тестирование элемента управления

Поскольку элемент управления DemoCalculator наследуется от класса **UserControl**, его поведение можно проверить с помощью Контейнера для тестирования пользовательских элементов управления.

6.1. Нажмите клавишу **F5**, чтобы собрать и запустить элемент управления DemoCalculator в **Контейнере для тестирования пользовательских элементов управления**.

6.2. Выберите границу между панелями **Splitcontainer** и перетащите ее влево и вправо. Размеры элемента **TableLayoutPanel** и всех его дочерних элементов управления будут изменяться в соответствии с доступным пространством.

6.3. Завершив тестирование элемента управления, нажмите кнопку **Заккрыть**.

9.2. Задания для самостоятельной работы

Практическая работа 7. Создать калькулятор, аналогичный калькулятору Windows, вид: обычный.

Практическая работа 8. Бег. Пользователь указывает количество км за каждый день в течение некоторого периода времени. Программа находит общий пробег, среднее значение в день, наименьшее и наибольшее значения.

Практическая работа 9. Создать приложение "Ежедневник". Возможности: создание записей двух видов: задача и заметка. Для задачи указывается срок выполнения. Заметка относится к одной из категорий. Список категорий можно изменять. Добавить поиск по содержанию заметок/задач в рамках выбранных категорий.

АННОТАЦИЯ ДИСЦИПЛИНЫ

Цель дисциплины - приобретение знаний, навыков и умений в области программирования аппаратных интерфейсов и интерфейсов пользователя, овладение знаниями и навыками в программировании на языках высокого уровня, основами дизайна, эргономики и юзабилити пользовательских интерфейсов.

Задачи дисциплины:

1. Изучение архитектур пользовательских и аппаратных интерфейсов;
2. Изучение основ дизайна и эргономики интерфейсов;
3. Практическая разработка интерфейсов CUI средствами C++ и Java.

В результате освоения дисциплины обучающийся должен:

Знать: технологии и методы программирования с использованием IDE в задачах разработки интерфейсов на языках Java и C++; методы принципы и средства проектирования графических интерфейсов пользователя (GUI), методы тестирования программных компонентов пользовательских интерфейсов на основе реализации юзабилити для потенциальных пользователей.

Уметь: применять технологии и методы программирования с использованием IDE в задачах разработки интерфейсов GUI; использовать оптимальные принципы и методы проектирования интерфейсов в задачах информатизации гуманитарной сферы; тестировать компоненты программного обеспечения информационных систем на основе юзабилити человеко-компьютерного взаимодействия.

Владеть: технологиями и методами программирования с использованием IDE в задачах разработки интерфейсов на языках Java и C++; методами принципами и средствами проектирования графических интерфейсов пользователя (GUI) на основе достижений эргономики и дизайна; практическими навыками проектирования и тестирования пользовательских интерфейсов.