

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (Ф) СПбГУТ)**

**Составил
Ю.С. Маломан**

**УП.01 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ
(часть 2)**

Методические указания по выполнению практических работ

по специальности:

09.02.07 – Информационные системы и программирование

Архангельск 2022

Маломан, Ю.С. УП.01 Разработка программных модулей (часть 2). Методические указания по выполнению практических работ. – Архангельск: АКТ (ф) СПбГУТ, 2022.

Методические указания по выполнению практических работ содержат описания работ, предусмотренных рабочей программой по УП.01 «Разработка программных модулей» (часть 2). Каждая работа рассчитана на 2 часа, общий объём составляет 54 часа. Методические указания по выполнению практических работ предназначены для студентов очной формы обучения по специальности 09.02.07 – Информационные системы и программирование.

Рассмотрено и одобрено на заседании цикловой комиссии Информационных технологий и математических дисциплин Архангельского колледжа телекоммуникаций (филиал) СПбГУТ.

Содержание

Практическая работа №1. Разработка линейных алгоритмов на Python	4
Практическая работа №2. Разработка разветвляющихся алгоритмов на Python	6
Практическая работа №3. Разработка циклических алгоритмов на Python	8
Практическая работа №4. Обработка списков на Python	10
Практическая работа №5. Обработка строк на Python	12
Практическая работа №6. Обработка исключений на Python	14
Практическая работа №7. Разработка функций на Python	16
Практическая работа №8. Создание и использование модулей на Python	18
Практическая работа №9. Разработка классов на Python	20
Практическая работа №10. Тестирование и отладка приложений на Python	22
Практическая работа №11. Обработка строковых данных с помощью регулярных выражений в Python	25
Практическая работа №12. Работа с файлами и каталогами в Python	27
Практическая работа №13. Работа с данными различных форматов на Python	29
Практическая работа №14. Разработка приложений с графическим интерфейсом пользователя на Python	30
Практическая работа №15. Обработка событий в оконном приложении на Python	33
Практическая работа №16. Разработка сетевых приложений на Python	35
Практическая работа №17. Интеграция Python с другими языками программирования	39
Практическая работа №18. Разработка многопоточных приложений на Python	40
Практическая работа №19. Настройка интерфейса приложения	41
Практическая работа №20. Использование текстовых элементов управления	44
Практическая работа №21. Навигация по страницам	47
Практическая работа №22. Привязка к данным	49
Практическая работа №23. Работа с табличными данными	51
Практическая работа №24. Работа с мультимедиа	54
Практическая работа №25. Применение ресурсов и стилей	56
Практическая работа №26. Работа с БД	58
Практическая работа №27. Взаимодействие с сетью	60

Практическая работа №1

Разработка линейных алгоритмов на Python

1 Цель работы

- 1.1 Научиться разрабатывать линейные алгоритмы на Python;
- 1.2 Научиться считывать введенные пользователем данные;
- 1.3 Научиться выводить данные в форматированном виде;
- 1.4 Закрепить навык составления программ методами структурного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.1-3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Запросить у пользователя ввод двух целых чисел a и b . Реализовать вывод результата выполнения арифметических операций (сложение, вычитание, умножение, деление, получение целой части от деления, остаток от деления, возведение в степень), минимальное и максимальное из чисел. Дополнительные переменные не использовать.

Пример (при $a = 6$, $b=4$):

$$a + b = 10$$

$$a - b = 2$$

$$a * b = 24$$

...

$$\min = 4$$

$$\max = 6$$

- 5.2 Изучить варианты форматирования строк с использованием функции `format`. Реализовать вывод для задания 2 в следующем виде, используя один вызов функции `format()` (например, при $a = 6$, $b=4$):

$$6 + 4 = 10$$

$$6 - 4 = 2$$

$$6 * 4 = 24$$

...

5.3 Запросить у пользователя ввод двух чисел *a* и *b*. Реализовать вывод случайного вещественного числа в диапазоне от *a* до *b*, подключив модуль `random`.

5.4 Запросить у пользователя ввод целого числа. Реализовать вывод этого числа в двоичной, восьмеричной, десятичной и шестнадцатеричной системах счисления различными способами (с использованием форматирования строк и специализированных методов).

5.5 Запросить у пользователя ввод вещественного числа, реализовать его округление:

- до трех знаков после запятой, используя метод `format`,
- до *N* знаков после запятой, используя метод `round` (*N* должно вводиться пользователем).

5.6 Запросить у пользователя ввод года, реализовать вывод `True` или `False` в зависимости от того, високосный год введен или нет (год является високосным в двух случаях: либо он кратен 4, но при этом не кратен 100, либо кратен 400).

5.7 Написать программу, обменивающую значения двух целых чисел (значения чисел вводятся с клавиатуры пользователем).

5.8 Запросить у пользователя ввод целого числа *n* (количество секунд, прошедшее с начала суток). Определите, сколько часов, минут и секунд будет показано на табло электронных часов, результат вывести в формате `чч:мм:сс`. Учесть, что число *n* может быть больше, чем количество секунд в сутках, но результат должен корректно выводиться (для этого отбрасывать количество секунд в сутках).

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Каким образом можно считать информацию с клавиатуры в приложении на Python?

8.2 Какое количество переменных одновременно может быть считано с клавиатуры?

8.3 Какой модуль в Python содержит математические функции и константы?

8.4 Как преобразовать считанную строку в простой тип данных в Python?

8.5 Как указать escape-символы в строке форматирования в Python?

Практическая работа №2

Разработка разветвляющихся алгоритмов на Python

1 Цель работы

- 1.1 Научиться применять операторы if-else, switch, тернарный оператор в программах на Python;
- 1.2 Закрепить навык составления программ методами структурного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.4.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Написать программу, вычисляющую значение функции $y(x)$ с округлением до трех знаков после запятой:

- $y(x) = \pi \cdot x^2$, если $x < -10$;
- $y(x) = x^4$, если $-10 \leq x < -5$;
- $y(x) = e^{|x|}$, если $-5 \leq x < 10$;
- $y(x) = 1/\sin(\sqrt{x})$, если $x \geq 10$.

- 5.2 Запросить у пользователя ввод трех чисел. Вывести на экран то из них, которое находится между двумя другими. При выполнении использовать цепочки сравнения и тернарный оператор (трехместный if).

- 5.3 Написать программу, определяющую по году и номеру месяца, сколько дней в месяце (с учетом високосных лет). Год является високосным, если он кратен 4 и не кратен 100 или если он кратен 400. При выполнении использовать тернарный оператор (трехместный if).

- 5.4 Запросить у пользователя ввод трех чисел. Вывести на экран информацию о том, можно ли построить треугольник со сторонами указанной длины и его тип (прямоугольный, равнобедренный, равносторонний), если построение возможно.

- 5.5 Запросить у пользователя два целых числа и битовую операцию (и, или, исключающее или, сдвиг влево, сдвиг вправо), исходные числа и результат выполнения операции в двоичном виде вывести на экран.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 В каких случаях применяется тернарный оператор?

8.2 В чем отличие между операторами if-else и switch?

8.3 В каких случаях целесообразно заменять оператор if-else оператором switch?

8.4 В каких случаях выполняется блок default оператора switch?

8.5 Как указать в операторе выбора выполнение одних и тех же действий для различных параметров, не дублируя код?

8.6 Каков формат тернарного оператора?

8.7 Является ли обязательной часть else условного оператора?

8.8 Какие логические операторы могут применяться при составлении условий?

Практическая работа №3

Разработка циклических алгоритмов на Python

1 Цель работы

- 1.1 Научиться разрабатывать и применять циклы в программах на Python;
- 1.2 Закрепить навык составления программ методами структурного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.5.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Написать программу, генерирующую случайное целое число от 1 до 1000 и выводящую на экран само число и ответ, является ли оно простым. При выполнении использовать цикл for.

5.2 Написать программу, загадывающую случайное число от 1 до 10. Пользователь пытается угадать загаданное число. Если угадал, сообщить, что он молодец. Если ввел число меньше или больше загаданного, вывести на экран соответствующую надпись и требовать повторный ввод.

5.3 Написать программу для перевода температуры по Цельсию в температуру по Фаренгейту от 100 до 0 градусов с шагом 10. Формула перевода: $F = C \cdot 1.8 + 32$. При выполнении использовать цикл for.

5.4 Написать программу, запрашивающую сумму покупки и внесенную покупателем сумму. Если сумма покупки меньше или равна 0, требовать повторить ввод суммы покупки. Если внесенная сумма меньше суммы покупки, сообщить об этом, указав недостающую сумму, и потребовать повторить ввод.

После корректного ввода исходных данных, если сдача не требуется, выводить на экран «Спасибо!»; если внесена сумма больше необходимой – «Возьмите сдачу» и указывать сумму сдачи.

5.5 Вывести N значений функции $y(x) = ax + b$ на отрезке от x_1 до x_2 (N, a, b, x_1 и x_2 вводятся пользователем. Все числа кроме N – вещественные).

Если x_1 больше x_2 , то значения функции выводить по убыванию, иначе – по возрастанию. Результат округлять до трех знаков после запятой. Значения функции на концах отрезка ($y(x_1)$ и $y(x_2)$) также должны быть выведены.

Образец вывода: $y(0.16) = 2*0.16+5=5.320$

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие виды циклических операторов имеются в Python?

8.2 В каком случае выполнится тело цикла с предусловием?

8.3 В каком случае выполнится тело цикла со счетчиком?

8.4 Для чего в циклах применяется оператор continue?

8.5 Для чего в циклах применяется оператор break?

8.6 Какое минимальное количество раз выполняется каждый из циклических операторов?

8.7 Какие операторы досрочного выхода из цикла имеются в Python?

8.8 В каком случае выполнится блок else в циклах на Python?

Практическая работа №4

Обработка списков на Python

1 Цель работы

- 1.1 Научиться создавать и обрабатывать списки в программах на языке Python;
- 1.2 Научиться применять стандартные методы класса List и разрабатывать собственные методы для обработки списков в программах на языке Python.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.6.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Запросить у пользователя количество элементов списка. Заполнить список случайными целыми числами от 0 до 100. Построчно вывести каждый элемент списка с указанием его индекса.

5.2 Запросить у пользователя количество элементов списка. Создать новый список, который должен заполнить пользователь. Вывести весь список одной строкой.

5.3 Дописать в один из списков все четные элементы из другого списка. Вывести полученный результат на экран одной строкой, разделенной пробелами (для этого в цикле использовать `print(список[i], end = " ")`)

5.4 Вывести на экран список, отсортированный по убыванию, не используя циклы.

5.5 Запросить у пользователя ввод числа. Вывести количество совпадений со значениями элементов в списке и удалить все совпадения из списка, не используя циклы.

5.6 Реализовать вставку в существующий и заполненный список *n* элементов. *n*, индексы и значения указываются пользователем. Вывести полученный результат на экран.

5.7 Создать словарь, содержащий код и название книги. Количество элементов, ключи и значения вводятся пользователем.

5.8 Для созданного словаря вывести на экран список ключей и список значений словаря. Запросить у пользователя ключ и вывести значение словаря, соответствующее указанному ключу.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «список»?

8.2 В чем отличие списков и словарей в Python?

8.3 Как обратиться к элементу списка в Python?

8.4 В чем отличие списков от массивов в Python?

Практическая работа №5

Обработка строк на Python

1 Цель работы

1.1 Научиться создавать и обрабатывать строковые данные в программах на языке Python;

1.2 Научиться применять стандартные методы обработки строк и разрабатывать собственные методы для обработки строк в программах на языке Python.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.7.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Запросить у пользователя строку:

- вывести строку на экран 5 раз, не используя циклы;
- вывести длину строки и каждый символ строки с указанием его индекса.
- вывести каждый второй символ строки на экран с указанием его позиции в строке (не индекса).

5.2 Запросить у пользователя строку и два числа (начальная и конечная позиции), вывести все символы находящиеся в диапазоне между двумя указанными позициями.

5.3 Запросить у пользователя строку. Изменить первый и последний символы строки на #.

5.4 Запросить у пользователя строку и вывести на экран информацию о том, состоит ли строка только из цифр, только из букв, только из цифр и букв. Если строка состоит только из букв дополнительно сообщить, только ли из строчных или прописных букв она состоит.

5.5 Запросить у пользователя строку, разделить ее на набор подстрок (разделитель: пробел). Объединить в новую строку все слова, встречающиеся в исходной строке, через запятую.

5.6 Запросить у пользователя две строки. Определить, сколько раз встречается первая строка во второй и вывести на экран номера позиций, с которых она встречается.

5.7 Написать программу, определяющую, является ли введенное слово палиндромом (т.е. одинаково читается справа налево и слева направо). Регистр символов не должен учитываться.

5.8 Написать программу, заменяющую все двойные пробелы в строке на одинарные (в результате выполнения программы в строке не должно остаться двух и более подряд идущих пробелов).

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 К какому типу переменных относятся переменные типа string?

8.2 Какая максимальная длина строки допустима в Python?

8.3 Какие операции допустимы над строковыми данными?

8.4 Каков синтаксис объявления «сырых» строк?

Практическая работа №6

Обработка исключений на Python

1 Цель работы

- 1.1 Научиться перехватывать и обрабатывать исключения в программах на Python;
- 1.2 Научиться настраивать и использовать объекты класса Exception в программах на Python.

2 Литература

- 2.1 Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=395908>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – п.3.7.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Разработать приложение, позволяющее найти частное чисел а и b. Числа вводятся пользователем. Добавить перехват исключения, происходящего при попытке деления на 0. Результат деления или сообщение о невозможности операции (в зависимости от введенных данных) вывести на экран.

5.2 Разработать приложение, позволяющее найти частное чисел а и b. Числа вводятся пользователем. Добавить перехват исключения, происходящего при попытке деления на 0. В случае, если пользователь ввел 0, сообщить, что 0 вводить нельзя, и требовать повторного ввода числа b, после того, как будет получено корректное значение b в блоке finally выводить результат деления.

5.3 Разработать приложение, позволяющее корректно вычислить значение следующего выражения:

$$\sqrt{(x+y+z)/(x-y+z)^2}$$

Числа вводятся пользователем. Добавить два перехватчика исключений (ввод нечисловых данных и попытка деления на 0).

5.4 Модифицировать приложение, разработанное в п.3, так, чтобы в случае, когда под корнем отрицательное число, при помощи raise генерировалось исключение.

Пример генерации своего исключения:

try:

```
raise ValueError("Описание исключения")
except ValueError as msg:
    print(msg) #Выведет: Описание исключения
```

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «исключение»?

8.2 Каков синтаксис перехватчика исключений в Python?

8.3 Какие классы являются дочерними для класса Exception?

Практическая работа №7

Разработка функций на Python

1 Цель работы

- 1.1 Научиться разрабатывать на Python программы, содержащие пользовательские функции (подпрограммы);
- 1.2 Закрепить навык составления программ методами процедурного и структурного программирования.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – п.9.1.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Написать и протестировать функцию, возводящую переданное в параметрах число a в степень x . По умолчанию a должно быть равно 2. При реализации можно использовать `**`.

Функцию снабдить комментариями согласно следующему шаблону (описание параметров дано в сигнатуре функции):

```
def названиеФункции(параметр1: 'описание параметра 1' = значение1,
параметр2: 'описание параметра 2' = значение2):
    """Комментарий к функции
    """
    тело функции
```

5.2 Написать и протестировать функцию, рекурсивно вычисляющую факториал числа. Если исходные данные некорректны (например, передана строка или отрицательное число), возвращать -1. Стандартную функцию вычисления факториала не использовать.

5.3 Написать и протестировать функцию, выводящую на экран сумму, среднее, максимум, минимум и количество всех чисел, переданных через параметры.

5.4 Написать и протестировать функцию, изменяющую значения переданного в нее списка путем умножения каждого из элементов списка на переданное в параметрах число. Если второй параметр отсутствует, умножать на -1.

5.5 Написать и протестировать лямбда-функцию, вычисляющую значение $y=a*x+b$.

6 Порядок выполнения работы

6.1 Запустить Python IDLE, создать и протестировать в одном файле набор функций согласно заданию из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какие способы передачи параметров в функцию существуют в Python?

8.2 В какой части программы можно объявлять функции?

8.3 Как задать параметры по умолчанию в функциях на Python?

8.4 Что такое «рекурсия»?

8.5 Какие проблемы могут возникать при реализации рекурсивных алгоритмов на электронных вычислительных машинах?

8.6 В каких случаях оправдано применение рекурсивных функций?

Практическая работа №8

Создание и использование модулей на Python

1 Цель работы

- 1.1 Научиться разрабатывать модули и использовать их в программах на Python;
- 1.2 Закрепить навык составления программ методами модульного и процедурного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – п.9.2.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Требуется создать модуль, содержащий следующие функции:

- 5.1 Функцию `hello`, выводящую «Hello, World» или «Hello, имя», если в нее передан параметр.
- 5.2 Функцию `arithmetic`, принимающую 3 аргумента: первые два - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент `+`, сложить их; если `-`, то вычесть; `*` - умножить; `/` - разделить (первое на второе). В остальных случаях вернуть строку «Неизвестная операция».
- 5.3 Функцию `square`, принимающую 1 аргумент – сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.
- 5.4 Функцию `season`, принимающую 1 аргумент – номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень).
- 5.5 Функцию `bank`, принимающая аргументы `a` и `years`, и возвращающую сумму, которая будет на счету пользователя. Пользователь делает вклад в размере `a` рублей сроком на `years` лет под 10% годовых (каждый год размер его вклада увеличивается на 10%. Эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты).

6 Порядок выполнения работы

- 6.1 Запустить Python IDLE и выполнить все задания из п.5.

Создать и протестировать в отдельной программе свой модуль, содержащий реализацию набора функций. До подключения модуля к отдельной программе можно проводить тестирование модуля следующим образом:

```
def hello():  
    print('Hello, world!')  
  
if __name__ == "__main__": # если запущен модуль, выполнить функцию hello()  
    hello()
```

Все функции снабдить многострочными комментариями, описывающими, что делает функция, и описанием параметров на русском языке согласно следующему шаблону (описание параметров дано в комментарии функции):

```
def названиеФункции(параметр1, параметр2):  
    """Комментарий к функции  
  
    параметр1 - описание параметра 1 (default: значениеПоУмолчанию1)  
    параметр2 - описание параметра 2 (default: значениеПоУмолчанию2)  
    """  
  
    тело функции
```

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «модуль»?

8.2 Как подключить модуль к приложению на Python?

8.3 Как подключить определенный набор функций модуля к приложению на Python?

8.4 В чем преимущества модульного подхода к разработке приложений?

Практическая работа №9

Разработка классов на Python

1 Цель работы

- 1.1 Научиться создавать классы в программах на Python;
- 1.2 Закрепить навык составления программ методами модульного и объектно-ориентированного программирования.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.11.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать класс Автор с полями ФИО, страна. Добавить в класс метод, выводящий на экран информацию об авторе. В основной программе создать список из N авторов. Исходные данные вводятся пользователем. Вывести список всех авторов и список только русских авторов на экран.

5.2 Создать класс Книга у которого должны быть:

- приватное поле содержание (список произведений книги, который должен быть изначально пустым);
- конструктор `__init__`, который создает поле названиеКниги, меняет его значение на значение, переданное в параметрах конструктора, и выводит на экран сообщение «книга название создана»;
- деструктор `__del__`, который выводит на экран сообщение «книга название удалена».

5.3 В классе книга создать следующие методы:

- метод, добавляющий переданное в параметрах название произведения в содержание книги;
- метод, возвращающий количество произведений, имеющихся в книге.

5.4 В классе Книга создать метод, выводящий на экран информацию о книге в следующей форме:

Книга: название книги

Содержание:

- 1) название1
- 2) название2

...

5.5 Создать класс КнигаАвтора, являющийся наследником классов Автор и Книга. Добавить в класс метод, выводящий на экран ФИО автора и название книги и содержание книги.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5. При разработке классов документировать их (снабдить классы и методы классов многострочными комментариями).

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «наследование» в контексте объектно-ориентированного программирования?

8.2 Что такое «полиморфизм» в контексте объектно-ориентированного программирования?

8.3 Как реализуется принцип полиморфизма при разработке классов Python?

8.4 Что такое «инкапсуляция» в контексте объектно-ориентированного программирования?

8.5 Как реализуется принцип инкапсуляция при разработке классов Python?

8.6 Для чего применяются конструкторы классов?

8.7 В чем преимущества применения дочерних классов?

Практическая работа №10

Тестирование и отладка приложений на Python

1 Цель работы

- 1.1 Сформировать умение составлять полные и избыточные наборы тестов;
- 1.2 Научиться проводить тестирование и отладку приложений на Python;
- 1.3 Научиться применять интерактивный отладчик pdb;
- 1.4 Закрепить навык разработки и тестирования приложений на языке Python.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.9.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Составить тестовые наборы данных для скриптов, разработанных на практическом занятии №2. Оформить тестовые наборы в виде таблиц.

5.2 Запуск пошаговой отладки в Python IDLE

5.2.1 Написать скрипт, в котором пользователь вводит число n. После ввода скрипт должен выполнять вычисление факториала числа n и выводить результат на экран.

```
n = int(input("n="))
fact = 1
for x in range(2, n+1):
    fact = fact*x
print(fact)
```

5.2.2 Включить отладчик в Python Shell (меню Debug – Debugger). После включения откроется окно Debug Control, в интерпретаторе появится сообщение о включении отладчика [DEBUG ON] (при закрытии отладчика появится сообщение [DEBUG OFF]).

В окне Debug Control отметить все флажки.

5.2.3 Запустить скрипт из п.5.2.1 на пошаговое выполнение

Для того, чтобы выполнять пошаговую отладку без захода в функции (стандартных функций int() и input()), нажать в отладчике Over.

Для того, чтобы выйти из функций, нажать в отладчике Out.

В ходе отладки изучить изменение значений переменных в окне отладчика.

5.3 Создание и применение брейкпоинтов в Python IDLE

5.3.1 Включить отладчик в Python Shell (меню Debug – Debugger) и запустить скрипт из п.5.2.1.

5.3.2 Добавить в скрипт на строке `fact = fact*x` брейкпоинт (ПКМ – Set Breakpoint)

5.3.3 Запустить скрипт из п.5.2.1 на пошаговое выполнение

Для того, чтобы переходить по брейкпоинтам, нажать в отладчике Go.

В ходе отладки изучить изменение значений переменных в окне отладчика.

5.4 Запуск пошаговой отладки с заходом в функции в Python IDLE

5.4.1 Изменить код скрипта из п.5.2.1, добавив в него функцию, вычисляющую в цикле значение факториала числа n. После описания функции выполнить ее вызов.

5.4.2 Включить отладчик в Python Shell (меню Debug – Debugger) и запустить скрипт из п.5.4.1.

5.4.3 Запустить скрипт из п.5.4.1 на пошаговое выполнение с заходом в функцию вычисления факториала.

Для того, чтобы заходить в функции, нажать в отладчике Step.

Для того, чтобы досрочно вернуться из функции, нажать в отладчике Out.

В ходе отладки изучить изменение значений переменных в окне отладчика.

5.5 Провести отладку приложения на языке Python, содержащего цикл для вычисления факториала, используя встроенный интерактивный отладчик `pdb`. Запуск отладчика осуществить через интерактивный интерпретатор.

Для подключения отладчика нужно импортировать `pdb`.

Для вызова отладчика в определенной строке программы нужно написать:
`pdb.set_trace()`

При достижении этой строки в интерпретаторе будет отображаться текущая строка скрипта и строка (Pdb), в которой можно задать настройки отладчика `pdb`.

Например, для вывода переменной, нужно написать ее название.

Для выполнения следующего шага требуется написать `step`, для выхода – `exit`.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое `pdb`?

8.2 Как просмотреть стек вызова?

- 8.3 Как задать пошаговое выполнение приложения, используя pdb?
- 8.4 Как добавить в приложение точки останова, используя pdb?
- 8.5 Как настроить точки останова pdb?
- 8.6 Что такое «отладка»?
- 8.7 Что такое «тестирование»?
- 8.8 Что входит в процесс отладки программ?
- 8.9 В каком виде могут быть представлены наборы тестов?

9 Приложение

Элементы окна Debug Control:

Кнопки:

- Go - выполняет остальную часть кода как обычно или пока не достигнет брейкпоинта (аналог в VS - F5).
- Step - шаг на одну строку кода. Если строка является вызовом функции, отладчик перейдет к функции (аналог в VS - F11).
- Over - шаг на одну строку кода. Если строка является вызовом функции, отладчик не перейдет к функции, а вместо этого перейдет к вызову (аналог в VS - F10).
- Out - продолжает переходить по строкам кода до тех пор, пока отладчик не выйдет из функции, в которой он находился при нажатии кнопки Out (аналог в VS - Shift+F11).
- Quit - немедленное завершение программы.

Флажки:

- Stack – отображать стек вызова (функции и содержащие их файлы)
- Source – отображать код программы с выделением текущей строки. Если файл, содержащий вызываемую функцию, закрыт, он будет открыт.
- Locals – отображать локальные переменные
- Globals – отображать глобальные переменные

Практическая работа №11

Обработка строковых данных с помощью регулярных выражений в Python

1 Цель работы

- 1.1 Научиться составлять регулярные выражения в программах на Python;
- 1.2 Научиться применять регулярные выражения для поиска и замены подстрок в программах на Python.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.7.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Написать регулярное выражение, разделяющее строку на предложения и вывести каждое предложение с новой строки. Признак конца предложения: точка, вопросительный знак, восклицательный знак, многоточие.

Для разделения строки на список использовать:

`re.split(r'паттерн', строка)`

- 5.2 Написать регулярное выражение, заменяющее слова «редиска» и фразу «нехороший человек» (и то и другое в любых склонениях и числе) на текст «*давайте жить дружно*».

Для замены подстроки на новую подстроку использовать:

`re.sub(r'паттерн', 'новая подстрока', строка)`

- 5.3 Написать регулярное выражение, выделяющее из строки все даты в формате дд.мм.гггг и выводящее их на экран. Вместо дд и мм может быть от 1 до 2 чисел, вместо гггг 2 или 4 числа. Месяц может быть от 1 до 12, день может быть от 1 до 31.

Для возврата списка всех совпадений использовать:

`re.findall(r'паттерн', строка)`

- 5.4 Написать регулярное выражение (или набор регулярных выражений), проверяющее, является ли введенная строка надежным паролем. Требования к паролю: длина от 6 символов, обязательно наличие цифры, строчной и прописной буквы латинского алфавита. Если требования не выполнены, запрашивать повторный ввод до тех пор, пока пароль не будет соответствовать требованиям.

Для определения, есть ли в строке определенный текст, использовать:

`re.search(r'паттерн', строка)`

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «регулярное выражение»?

8.2 Для чего используются регулярные выражения?

8.3 Какие методы применяются в Python для работы с регулярными выражениями?

8.4 Как задать количество повторений символов в регулярных выражениях?

8.5 Как задать набор символов в регулярных выражениях?

8.6 Как привязать поиск к началу и концу строки в регулярном выражении?

8.7 Как задать буквенные, числовые и пробельные символы в регулярном выражении?

Практическая работа №12

Работа с файлами и каталогами в Python

1 Цель работы

- 1.1 Научиться работать с файлами и каталогами в программах на Python;
- 1.2 Закрепить навык составления программ методами процедурного и структурного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.10.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

В работе обязательно выполнять проверку существования файлов и папок и обрабатывать исключения.

5.1 Написать программу, выводящую на экран содержимое файла, название которого ввел пользователь. Если название файла заканчивается на .ру, после вывода его содержимого выполнять скрипт, содержащийся в этом файле, используя функцию `exec(текст скрипта)`.

Для получения расширения использовать:

`os.path.splitext(path)` - разбивает путь на пару (root, ext), где ext начинается с точки и содержит не более одной точки.

5.2 Написать программу, позволяющую записать введенный пользователем текст в файл (каждая введенная строка должна записываться с новой строки в конец файла). Признак конца ввода – введенная пользователем строка `end`. Перед началом, если файл существует, запросить у пользователя информацию, нужно дописать или перезаписать файл.

5.3 Написать программу, запрашивающую у пользователя имя исходного файла и предлагающую три операции: вывод содержимого файла на экран, удаление файла, переименование файла. Если файл требуется переименовать, запрашивать новое имя файла.

5.4 Написать программу, удаляющую все файлы с расширением .txt в указанной пользователем директории и всех ее поддиректориях.

Для получения списка поддиректорий и файлов использовать:

`os.walk(каталог)` - возвращает кортеж (каталог, подкаталоги, файлы)

5.5 Написать программу, записывающую в файл список поддиректорий и файлов указанной папки. В файле у каждой поддиректории и файла указать их размер в килобайтах.

Для определения размера файлов использовать:

`os.path.getsize(path)` - размер файла в байтах.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего применяется модуль `os`?

8.2 Как рекурсивно перебрать все каталоги, находящиеся в указанном, в приложении на Python?

8.3 Какие существуют способы открытия файлов в программах на Python?

8.4 Какие операции над каталогами можно выполнять в программах на Python?

Практическая работа №13

Работа с данными различных форматов на Python

1 Цель работы

- 1.1 Научиться обрабатывать данные различных форматов в программах на Python;
- 1.2 Закрепить навык разработки и тестирования приложений на языке Python;
- 1.3 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.10.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Разработать приложение, позволяющее считывать и модифицировать информацию в xml-файле, используя пакет xml.
- 5.2 Разработать приложение, позволяющее создавать и модифицировать wav-файл, используя модуль wave.
- 5.3 Разработать приложение, позволяющее считывать и модифицировать информацию в таблицах базы данных SQLite, используя DB API.

6 Порядок выполнения работы

- 6.1 Запустить Python IDLE и выполнить все задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое DB API?
- 8.2 Какие модули входят в пакет xml?
- 8.3 Какие модули Python позволяют обрабатывать мультимедиа информацию?

Практическая работа №14

Разработка приложений с графическим интерфейсом пользователя на Python

1 Цель работы

- 1.1 Научиться создавать оконные приложения на Python, используя библиотеку Tkinter;
- 1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.12.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать оконное приложение «Форма авторизации» с однострочными полями Логин, Пароль, флажком «Запомнить пароль», подписями и кнопкой Авторизоваться. Настроить размер формы 200x300 и заголовок окна.

5.2 Создать оконное приложение «Форма регистрации» со следующими элементами и подписями:

- однострочные поля Логин, Пароль,
- многострочное поле О себе,
- переключатели для указания пола,
- список из пяти пунктов для выбора материка,
- кнопка Зарегистрироваться.

Настроить цвет фона формы и элементов управления и заголовок окна.

5.3 Создать оконное приложение, в котором связать виджеты поле ввода, флажок, переключатель с ассоциированными переменными-объектами (IntVar, StringVar и т.д.) и, используя textvariable, определить текст-содержимое или текст-надпись виджетов. Написать скрипт, в котором значения ассоциированных переменных должны отображаться в метке через запятую.

5.4 Создать приложение с меню, содержащим два пункта: Color и Size. Пункт Color должен содержать три команды (Red, Green и Blue), меняющие цвет рамки на главном окне. Пункт Size должен содержать две команды (500x500 и 700x400),

изменяющие размер рамки. Привязать к пунктам меню «горячие клавиши» и выполнение функций.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «виджет»?

8.2 Что содержит библиотека Tkinter?

8.3 Какие виджеты описаны в библиотеке Tkinter?

8.4 Какие свойства отвечают за внешний вид виджета?

8.5 Как разместить виджет на форме?

9 Приложение

Изучить справку: The Python Standard Library – Graphical User Interfaces with Tk

Для проверки работоспособности модуля tkinter можно выполнить следующий скрипт:

```
import tkinter
tkinter._test()
```

Способы создания и описания объектов оконного приложения:

1 вариант (процедурный подход):

```
from tkinter import *
```

```
def hello(event):
```

```
print ("Очередной 'Hello World!'") root = Tk()
```

```
# 1 способ создания кнопки
```

```
btn=Button(root,                                #родительское окно
            text="Hello",                        #надпись на кнопке
            width=30, height=5,                 #ширина и высота
            bg="white", fg="black")             #цвет фона и надписи
```

```
# 2 способ создания кнопки
```

```
btn = Button(root)
```

```
btn["text"] = "Hello"
```

```
btn["width"] = 30
```

```
btn["height"] = 5
```

```
btn["bg"] = "white"
btn["fg"] = "black"
```

```
# привязка кнопки к функции
btn.bind("<Button-1>", hello) #при нажатии на левую кнопку мыши вызывается
функция hello
```

```
# 3 способ создания кнопки (с привязкой кнопки к функции)
btn = Button(command = hello) #при нажатии на кнопку вызывается функция
hello (без параметров)
```

```
btn.pack() #расположить кнопку на главном окне
root.mainloop() # отображение главного окна. Команда д.б. последней в скрипте
2 вариант (объектно-ориентированный подход):
```

```
from tkinter import*
```

```
class Btn_hello:
    def init(self):
        self.btn = Button(root)
        self.btn["text"] = "Hello"
        self.btn.bind("<Button-1>", self.hello) self.btn.pack()
    def hello(self,event):
        print ("Очередной 'Hello World!'")
```

```
root = Tk()
obj = Btn_hello()
root.mainloop()
```


Практическая работа №15

Обработка событий в оконном приложении на Python

1 Цель работы

- 1.1 Научиться обрабатывать действия пользователя, события клавиатуры и мыши в программах на Python;
- 1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.12.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Написать оконное приложение, в которое добавить многострочное поле ввода и кнопку Сохранить. При нажатии на кнопку или клавиши Ctrl-S должно происходить открытие окна, позволяющего сохранить текст в файл. При нажатии Esc форма должна закрываться.

5.2 Написать оконное приложение с тремя полями ввода. При нажатии на любое из полей ввода левой кнопкой мыши в метку на форме должно выводиться название активного поля ввода, при нажатии правой – в консоль. Для реализации подписки на событие использовать `root.bind_class`.

5.3 Написать оконное приложение, на котором разместить метку с подписью «Координаты мыши». При движении мыши выводить в метку координаты x и y мыши (для этого в обработчике движения мыши есть свойства `event.x`, `event.y`).

5.4 Написать оконное приложение, на котором разместить метку с подписью «Нажатые клавиши». При нажатии любой из клавиш добавлять в метку последний нажатый символ (для этого в обработчике `Key` есть свойства `event.char`).

6 Порядок выполнения работы

- 6.1 Запустить Python IDLE и выполнить все задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какова общая форма написания обработчиков событий?

8.2 Как добавить прослушиватель на события мыши?

8.3 Как добавить прослушиватель на события клавиатуры?

8.4 Какие виды событий мыши можно обработать в программах на Python?

8.5 Какие виды событий клавиатуры можно обработать в программах на Python?

Практическая работа №16

Разработка сетевых приложений на Python

1 Цель работы

- 1.1 Научиться создавать сетевые клиент-серверные приложения на Python;
- 1.2 Закрепить навык составления программ методами процедурного и событийно-ориентированного программирования.

2 Литература

- 2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.12.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание клиент-серверного сетевого приложения

- 5.1.1 Создание скрипта сервера

Код сервера, принимающего подключение по порту 50007

и пересылающего полученные данные отправителю

```
import socket
```

```
HOST = "                # Слушать все интерфейсы
```

```
PORT = 50007            # Порт, который необходимо слушать
```

```
# Создание интернет-сокета для использования по протоколу TCP
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
# прослушивание всех интерфейсов на порту PORT
```

```
s.bind((HOST, PORT))
```

```
# Перевод сокета в режим ожидания. Параметр – количество одновременных
```

```
# соединений (1 означает, что может быть одно соединение одновременно)
```

```
s.listen(1)
```

```
# ожидание соединения от клиента
```

```
conn, addr = s.accept()
```

```
print('Подключен клиент: ', addr) # addr – ip и номер порта клиента
```

```
# в цикле выполняется чтение данных от клиента
```

```
while True:
```

```

# Считывание данных от клиента по 1024 байта
data = conn.recv(1024)
if not data:
    break
# вывод текста в кодировке UTF-8
print('Получено сообщение:', data.decode('utf-8'))
# Отправка считанных данных обратно клиенту
conn.sendall(data)

# Закрытие соединения
conn.close()

```

5.1.2 Создание скрипта клиента

```

# Код клиента, подключающегося к локальному серверу по порту 50007
# и пересылающего полученные данные отправителю
import socket
HOST = 'localhost'      # Имя сервера, к которому требуется подключиться
PORT = 50007            # Номер порта на сервере

# Создание интернет-сокета для использования по протоколу TCP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Соединение с сервером
s.connect((HOST, PORT))

# отправка серверу байтовой строки
# (для преобразования в байты используется функция encode())
s.sendall('Hello, world'.encode())
# Прием от сервера данных
data = s.recv(1024)
# Закрытие соединения
s.close()
# Вывод на экран полученных данных
print('Сообщение сервера: ', repr(data))

```

5.1.3 Запустить для каждого скрипта свою копию IDLE (нужно для того, чтобы они работали в различных потоках) и осуществить сначала запуск сервера, потом запуск клиента.

5.2 Создание общего чата

5.2.1 Изменить код клиента следующим образом:

- убрать вывод строки, полученной с сервера;
- в бесконечном цикле запрашивать у пользователя ввод сообщения, которое должно отправляться на сервер;
- при вводе клиентом строки `end` работа клиента завершается.

5.2.2 Изменить код сервера следующим образом:

- реализовать постоянное прослушивание подключений (для этого добавить после `s.listen(1)` бесконечный цикл и перенести в него весь код, который написан после `s.listen(1)`);

- убрать пересылку сообщения клиенту и вывод сообщения о том, что клиент подключился к серверу;

- реализовать вывод в командной строке полученных данных следующим образом "дата и время (адрес клиента): сообщение".

Для вывода даты и времени использовать следующий код:

```
from datetime import datetime
```

```
datetime.strftime(datetime.now(), "%Y.%m.%d %H:%M:%S")
```

5.2.3 Протестировать, запустив сервер и несколько клиентов

5.3 Создание общего чата с информированием о том, какие пользователи подключились и какой пользователь отправил сообщение

5.3.1 Изменить код клиента следующим образом:

- при запуске клиента запрашивать логин и отправлять его на сервер при подключении;

- в бесконечном цикле запрашивать у пользователя ввод сообщения и отправлять логин и сообщение серверу.

5.3.2 Изменить код сервера следующим образом:

- при подключении клиента должно выводиться сообщение «подключился логин» (логин д.б. тот, что указал клиент при подключении);

- реализовать вывод в командной строке полученных данных следующим образом "дата и время логин: сообщение".

5.3.3 Протестировать, запустив сервер и несколько клиентов

5.4 Создание оконного клиента

5.4.1 Создать оконное клиентское приложение с кнопкой "Отправить" и двумя полями ввода (логин и сообщение). При нажатии на кнопку "Отправить" приложение должно действовать аналогично клиенту из п.5.3.

5.4.2 Протестировать, запустив сервер из п.5.3 и несколько оконных клиентов

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «сервер»?

8.2 Что такое «клиент»?

- 8.3 Какой модуль применяется для разработки сетевых приложений на Python?
- 8.4 Как передать данные от клиента серверу в приложениях на Python?

Практическая работа №17

Интеграция Python с другими языками программирования

1 Цель работы

1.1 Научиться проводить интеграцию приложений, написанных на языках программирования С и Python;

1.2 Закрепить навык разработки приложений на С и Python.

2 Литература

2.1 Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=395908>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.1.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Подключить к программе на Python модуль, написанный на С. Протестировать их взаимодействие.

5.2 Подключить к программе на С модуль, написанный на Python. Протестировать их взаимодействие.

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Зачем проводится интеграция Python с другими языками программирования?

8.2 Какую библиотеку требуется подключить в приложении на С для использования модулей, написанных на Python?

8.3 Как вызвать функцию, написанную на С, из приложения, написанного на Python?

Практическая работа №18

Разработка многопоточных приложений на Python

1 Цель работы

- 1.1 Научиться создавать многопоточные приложения на Python;
- 1.2 Закрепить навык разработки и тестирования приложений на языке Python.

2 Литература

2.1 Жуков, Р. А. Язык программирования Python. Практикум : учебное пособие / Р. А. Жуков. – Москва : ФОРУМ : ИНФРА-М, 2022. - URL:<https://znanium.com/read?id=395908>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.3.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Разработать многопоточное приложение, содержащее семафоры.
- 5.2 Разработать многопоточное приложение, реализующее обмен сообщениями между потоками, используя класс Event.
- 5.3 Разработать многопоточное приложение, реализующее обмен сообщениями между потоками, используя класс Condition.
- 5.4 Разработать приложение с низкоуровневым доступом к потокам управления, используя модуль thread.

6 Порядок выполнения работы

- 6.1 Запустить Python IDLE и выполнить все задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «многопоточность»?
- 8.2 Какие средства реализации многопоточных приложений имеются в Python?
- 8.3 Для чего применяются семафоры в многопоточных приложениях?
- 8.4 Для чего используется модуль thread?

Практическая работа №19

Настройка интерфейса приложения

1 Цель работы

1.1 Научиться выполнять создание и настройку приложения на Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.1-3.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Создание приложения

Создать мобильное приложение типа Xamarin.Forms и назвать его HelloApp.

При создании:

- выбрать пустой проект (пустой шаблон, создающий проект с минимальной функциональностью)

- указать тип ОС Android:

Запустить созданный проект на телефоне с ОС Android.

5.2 Добавление страниц

5.2.1 Добавление страницы и разметки с помощью кода

Добавить в приложение класс кода, в котором описать новую страницу Page21.

На странице разместить метку с текстом «Изучаем работу с Xamarin» и свои фамилии. Реализовать запуск этой страницы, изменив класс App

5.2.2 Добавление страницы с помощью шаблона Content Page (Страница содержимого) и разметки

Добавить в приложение новую страницу Page22 на основе шаблона.

На странице разместить метку с текстом «Изучаем работу с Xamarin» и свои фамилии. Реализовать запуск этой страницы, изменив класс App

5.3 Разметка XAML

Добавить в приложение новую страницу Page3 на основе шаблона. Реализовать запуск этой страницы, изменив класс App

Добавить на странице Page2 в XAML в StackLayout:

- метку Label, выровнять ее по центру, указать цвет текста зеленый, размер текста - 40, используя атрибуты XAML.

- метку Label и, используя свойство FormattedText, настроить в метке 3 блока текста (тэги Span), у каждого Span указать свои настройки текста, используя атрибуты XAML.

- метку Label и настроить размер и цвет текста, цвет фона, используя атрибуты XAML.

5.4 Взаимодействие XAML с C#

5.4.1 Добавить в приложение новую страницу Page41 на основе шаблона.

Реализовать запуск этой страницы, изменив класс App Добавить на страницу:

- кнопку с подписью «Нажми» и обработчиком события нажатия кнопки, - метку для отображения количества нажатий.

При нажатии на кнопку текст в метке должен отображать количество нажатий. Для реализации можно добавить в класс страницы поле count для хранения количества нажатий.

5.4.2 Добавить в приложение новую страницу Page42 на основе шаблона.

Реализовать запуск этой страницы, изменив класс App

Реализовать динамическую загрузку XAML (например, при нажатии на кнопку изменять текст в метке.

5.5 Применение контейнеров компоновки

5.5.1 Добавить в приложение новую страницу Page51 и расположить на ней по горизонтали набор меток с разным цветом фона и текста, используя StackLayout.

Реализовать запуск этой страницы, изменив класс App

5.5.2 Добавить в приложение новую страницу Page52 и расположить на ней по вертикали набор меток с разным цветом фона и текста, используя StackLayout и ScrollView. Размер текста указать таким, чтобы на экран вмещались только 2 метки.

Реализовать запуск этой страницы, изменив класс App

5.5.3 Добавить в приложение новую страницу Page53 и расположить на ней набор BoxView с разным цветом фона, используя AbsoluteLayout.

Реализовать запуск этой страницы, изменив класс App

5.5.4 Добавить в приложение новую страницу Page54 и расположить на ней набор BoxView с разным цветом фона, используя Grid (в сетке должно быть два столбца и три строки).

Реализовать запуск этой страницы, изменив класс App

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое XAML?

8.2 Что такое Xamarin Forms?

8.3 Как создать проект Xamarin Forms?

8.4 Какие контейнеры компоновки используются в приложениях Xamarin Forms?

Практическая работа №20

Использование текстовых элементов управления

1 Цель работы

1.1 Научиться использовать элементы управления Xamarin для ввода и вывода текстовых данных.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.4.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Применение Label

Добавить в приложение новую страницу Page1 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- метку Label

Реализовать отображение количества нажатий на метку в саму метку.

При четном количестве нажатий изменять цвет на зеленый, при нечетном – на желтый.

5.2 Применение Entry

Добавить в приложение новую страницу Page2 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- однострочное текстовое поле Entry для ввода логина

- однострочное текстовое поле Entry для ввода пароля (пароль должен быть скрытым)

- однострочное текстовое поле Entry для ввода email (указать соответствующее свойство Keyboard)

- однострочное текстовое поле Entry для ввода email (указать соответствующее свойство Keyboard)

- однострочное текстовое поле Entry для ввода телефона (указать соответствующее свойство Keyboard)

Поля ввода снабдить подсказками (placeholder)

Если в поле ввода логина не введен текст, отобразить метку с текстом красного цвета «Требуется ввести логин»

5.3 Применение Editor

Добавить в приложение новую страницу Page3 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- однострочное текстовое поле Entry для ввода логина
- многострочное текстовое поле Editor для ввода информации о себе

5.4 Применение Frame

Добавить в приложение новую страницу Page4 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Frame и указать для него цвет рамки и фона, скругление углов
- Label с текстом. Поместить Label во Frame

5.5 Применение DatePicker

Добавить в приложение новую страницу Page5 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Label с текстом «Выберите дату рождения»
- DatePicker для указания даты рождения (максимальное значение – текущая дата)
- Label с текстом «Выберите дату»
- DatePicker для указания выбранной даты (должна быть +/- неделя от текущей даты)

При изменении даты рождения вывести возраст человека

При изменении выбранной даты вывести на экран ее значение

5.6 Применение TimePicker

Добавить в приложение новую страницу Page6 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Label с текстом «Выберите время подъема»
- TimePicker для указания времени подъема

При изменении времени вывести на экран, сколько времени осталось до будильника.

5.7 Применение Picker

Добавить в приложение новую страницу Page7 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Label с текстом «Выберите время года»
- Picker для указания любимого времени года (указать времена для их отображения в виде списка).

При выборе времени года, совпадающего с текущим, вывести сообщение с поздравлением, при несовпадающем – сообщение о том, что пока придется подождать.

5.8 Применение Stepper

Добавить в приложение новую страницу Page8 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Stepper для указания любимого месяца (задать значения от 1 до 12 с шагом 1).

При выборе месяца, совпадающего с текущим, вывести сообщение с поздравлением, при несовпадающем – сообщение о том, что до нужного месяца осталось ждать N месяцев (вместо N – вычисленное значение).

5.9 Применение Slider

Добавить в приложение новую страницу Page9 на основе шаблона. Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- Slider для указания суммы в рублях (задать значения от 1 до 1000000, сменить цвета слайдера).

При изменении значения слайдера выводить на экран выбранное значение и сумму в долларах для выбранного количества рублей.

5.10 Применение Switch

Добавить в приложение новую страницу Page10 на основе шаблона.

Реализовать запуск этой страницы, изменив класс App Разместить на странице:

- метку с текстом.

- Switch для смены темы с темной на светлую.

При изменении значения переключателя изменять цвет текста и фона метки на противоположные.

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какой элемент управления предназначен для вывода текстовых данных?

8.2 Какой элемент управления предназначен для вывода текстовых данных?

8.3 Какой элемент управления предназначен для ввода даты и времени?

8.4 Какой элемент управления предназначен для ввода числовых данных?

Практическая работа №21

Навигация по страницам

1 Цель работы

1.1 Научиться выполнять навигацию по страницам приложения Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.9.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Переход по страницам и анимация переходов

Выполнить примеры со страницы <https://metanit.com/sharp/xamarin/5.1.php>

5.2 Работа со стеком навигации

Выполнить примеры со страницы <https://metanit.com/sharp/xamarin/5.2.php>

5.3 Передача данных между страницами

Выполнить примеры со страницы <https://metanit.com/sharp/xamarin/5.3.php>

Изменить примеры так, чтобы в приложении отображались не телефоны, а пользователи (с логином, паролем, возрастом)

Вместо PhonePage создать UserPage, отображающую информацию о выбранном пользователе.

5.4 Отправка сообщений

Выполнить примеры со страницы <https://metanit.com/sharp/xamarin/5.4.php>

5.5 Страница с вкладками

Выполнить примеры со страницы <https://metanit.com/sharp/xamarin/3.22.php>

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как добавить страницу в проект Xamarin Forms?

8.2 Как реализовать переход между страницами в Xamarin Forms?

8.3 Как передать данные между страницами в приложении Xamarin Forms?

8.4 Как добавить вкладки на страницу?

Практическая работа №22

Привязка к данным

1 Цель работы

1.1 Научиться выполнять привязку к данным в приложениях Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.7-8.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Привязка элементов управления к данным

Добавить на страницу поле ввода и две метки.

Реализовать привязку данных первой метки к данным, вводимым пользователем в поле ввода, используя BindingContext.

Реализовать привязку данных первой метки к данным, вводимым пользователем в поле ввода, используя Binding.

Добавить на страницу метку и слайдер для выбора месяца (от 1 до 12).

Реализовать привязку значения слайдера к метке и отформатировать значение, чтобы оно выводилось в следующем виде (вместо n должно подставляться выбранное значение): Вы выбрали n месяц

5.2 Смена режима привязки данных

Добавить на страницу два поля ввода и указать, что первое поле ввода связано со вторым полем ввода. Настроить одностороннюю привязку данных.

5.3 Применение конвертеров значений

Добавить на страницу метку и DatePicker. В DatePicker дата должна отображаться в полном формате, а в метке – в формате уууу-ММ-dd.

Реализовать привязку значения слайдера к метке и отформатировать значение, чтобы оно выводилось в следующем виде (вместо n должно подставляться выбранное значение): Вы выбрали n месяц

Добавить на страницу поле ввода для указания цвета и реализовать привязку цвета фона метки к цвету, указанному в поле ввода (например, пользователь указал Красный, и цвет фона метки соответственно изменился). Предусмотреть три цвета, если совпадения не найдено, сделать цвет белым.

5.4 Применение конвертеров значений

Добавить на страницу переключатель для смены темы на темную (по умолчанию – светлая) и реализовать привязку цвета фона и цвета текста метки к значению переключателя, создав два конвертера.

5.5 Реализация интерфейса INotifyPropertyChanged

Создать класс Пользователь со свойствами Логин, Пароль, Возраст.

Создать объект типа Пользователь и реализовать отображение данных пользователя в метках на странице.

Добавить на страницу две кнопки «Изменить возраст» (менять значение возраста на 18) и «Сбросить пароль» (менять значение пароля на qwerty).

Реализовать интерфейс INotifyPropertyChanged для оповещения меток об изменении значений свойств.

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «BindingContext»?

8.2 Как выполняется привязка к данным с помощью BindingContext?

8.3 Что такое «Binding»?

8.4 Как выполняется привязка к данным с помощью Binding?

8.5 Какие типы привязки данных поддерживаются элементами управления?

8.6 Для чего требуется реализовывать интерфейс INotifyPropertyChanged?

Практическая работа №23

Работа с табличными данными

1 Цель работы

1.1 Научиться выполнять привязку и отображение табличных данных в приложениях Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.8.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Вывод табличных данных

Создать в приложении массив или список строк, заполнить его названиями стран (Россия, США, Япония, другое)

Реализовать отображение на странице списка стран, используя ListView.

При нажатии на элемент списка отобразить в метке название выбранной страны.

5.2 Применение DataTemplate и ViewCell для настройки отображения элемента

Создать класс Пользователь со свойствами Логин, Пароль, Возраст.

Создать список типа Пользователь и реализовать отображение логина и возраста пользователей в ListView с использованием ViewCell.

Имя пользователя должно быть написано полужирным, возраст – синим цветом. Размер текста логина должен быть больше размера текста возраста.

Элементы должны отделяться чертой.

При нажатии на элемент списка отобразить логин выбранного пользователя вверху страницы, во всплывающем окне отобразить его пароль.

5.3 Применение DataTemplate и TextCell для настройки отображения элемента

Создать класс Пользователь со свойствами Логин, Пароль, Возраст.

Создать список типа Пользователь и реализовать отображение логина и пароля пользователей в ListView с использованием TextCell.

Имя пользователя должно быть написано одним цветом, пароль – другим цветом.

Элементы должны отделяться чертой.

5.4 Отображение изображений в ListView с использованием ImageCell

Добавить в проект набор изображений (фото пользователей) в папку Resources/Drawable.

Создать класс Пользователь со свойствами Логин, Пароль, Возраст, Фото.

Создать список типа Пользователь и реализовать отображение фото, логина и возраста пользователей в ListView с использованием ImageCell.

5.5 Отображение изображений в ListView с использованием ViewCell

Добавить в проект набор изображений (фото пользователей) в папку Resources/Drawable.

Создать класс Пользователь со свойствами Логин, Пароль, Возраст, Фото.

Создать список типа Пользователь и реализовать отображение фото, логина и возраста пользователей в ListView с использованием ViewCell.

5.6 Применение ObservableCollection для удаления

Создать класс Пользователь со свойствами Логин, Пароль, Возраст.

Создать ObservableCollection типа Пользователь и реализовать отображение логина, пароля и возраста пользователей в ListView с использованием ViewCell.

Элементы должны отделяться чертой.

Добавить на страницу кнопку Удалить и реализовать удаление выбранного пользователя из списка.

5.7 Применение ObservableCollection для вставки

Изменить приложение из п.6:

Добавить на страницу кнопку Добавить и поля ввода для указания логина, пароля и возраста.

Реализовать вставку нового пользователя в список. Данные нового пользователя брать из полей ввода.

5.8 Группировка данных

Создать класс Пользователь со свойствами Логин, Пароль, Роль.

Создать список типа Пользователь и реализовать отображение логина и возраста пользователей в ListView с использованием ViewCell.

Реализовать группировку по роли

Настроить ListView, чтобы вверху писалось «Список пользователей», внизу – текущая дата, сменить цвет разделителя, увеличить размер текста с названием группы.

При нажатии на элемент списка отобразить логин выбранного пользователя вверху страницы, во всплывающем окне отобразить его пароль.

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какой элемент управления используется для отображения данных в табличном виде в Xamarin Forms?

8.2 Для чего используется DataTemplate?

8.3 Для чего используется ViewCell?

8.4 Почему для отображения данных требуется использовать коллекцию типа ObservableCollection?

Практическая работа №24

Работа с мультимедиа

1 Цель работы

1.1 Научиться работать с мультимедиа-файлами в приложениях Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.4,11,18.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Работа с локальными изображениями

Добавить в приложение локальное изображение в папку Resources/drawable

Реализовать отображение локального изображения на странице, растянув его по странице.

5.2 Работа со встроенными изображениями

Добавить в приложение папку Images и добавить в нее изображение

Реализовать отображение встроенного изображения на странице, сохранив его пропорции.

5.3 Работа со встроенным браузером

Добавить на страницу кнопку и поле ввода для указания адреса сайта.

По умолчанию должна открываться страница с html-разметкой, код которой описан в приложении (на странице должен отображаться заголовок с текстом Добро пожаловать и список из трех пунктов).

При нажатии на кнопку – открывать в браузере страницу по указанному пользователем адресу (для просмотра нужно подключение телефона к сети).

5.4 Работа локальным HTML-файлом

Добавить в приложение html-страницу, css-файл и изображение для html-страницы.

Реализовать открытие локального html-файла во встроенном браузере.

5.5 Чтение и запись файлов

Разработать приложение, в которое добавить:

- поле ввода имени файла
- кнопку «Сохранить»

6 Порядок выполнения работы

- 6.1 Запустить Visual Studio и выполнить все задания из п.5.
- 6.2 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какой элемент управления позволяет отображать изображения в Xamarin Forms?
- 8.2 Для чего используется элемент управления WebView?
- 8.3 Какое пространство имен требуется подключить для работы с файловой системой?
- 8.4 В какой папке требуется хранить ресурсы мобильного приложения Xamarin Forms?

Практическая работа №25

Применение ресурсов и стилей

1 Цель работы

1.1 Научиться применять ресурсы и стили в приложении Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.6.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Добавление ресурсов объекта

5.1.1 Добавить в объект ResourceDictionary ресурсы, хранящие цвет фона, цвет текста и размер текста. Применить эти ресурсы к двум кнопкам на странице.

5.1.2 Добавить в объект ResourceDictionary страницы ресурс, хранящий цвет текста для разных платформ.

Применить ресурс к одной из кнопок на странице.

5.1.3 Сделать для одной из кнопок цвет фона динамическим ресурсом и реализовать изменение этого ресурса при нажатии на кнопку.

5.2 Добавление ресурсов приложения

Добавить в объект ResourceDictionary приложения ресурсы, хранящие цвет фона, цвет текста и размер текста.

Применить эти ресурсы к двум кнопкам на странице.

5.3 Добавление стилей

5.3.1 Добавить в объект ResourceDictionary страницы стиль, хранящий цвет фона, цвет текста и размер текста для всех кнопок. Применить стиль к двум кнопкам на странице.

5.3.2 Добавить в объект ResourceDictionary страницы стиль, основанный на стиле из 5.3.1, изменить в нем цвет фона кнопки.

Применить стиль к одной из кнопок на странице.

5.4 Добавление триггеров

Добавить в объект ResourceDictionary страницы стиль для всех полей ввода.

Описать в стиле триггер, изменяющий цвет фона и текста при фокусе в поле ввода. Применить стиль к одному из двух полей ввода на странице.

5.5 Добавление иконки приложения

Изменить стандартную иконку приложения на свою.

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Как добавить в проект мобильного приложения ресурсы?

8.2 Как описать стиль приложения?

8.3 Как настроить стиль приложения с помощью css?

8.4 Для чего используются триггеры свойств?

Практическая работа №26

Работа с БД

1 Цель работы

- 1.1 Научиться работать с БД SQLite, используя Xamarin.

2 Литература

- 2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.13.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Настройка проекта для работы с БД

Подключить в проект библиотеку sqlite-net-pcl для работы с SQLite

Создать интерфейс ISQLite с методом, возвращающим путь к БД

Создать класс SQLite_Android, реализующий интерфейс ISQLite

5.2 Описание класса таблицы пользователей

Описать класс User со следующими данными:

- Id (первичный ключ, автоинкремент)
- Login (уникальный)
- Password
- Role

5.3 Создание класса репозитория

5.3.1 Создать класс репозитория UserRepository, через который будет идти работа с данными

5.3.2 Внести изменения в файл App.xaml, чтобы подключение к БД было общим для всех

5.4 Создание страницы, выводящей записи из БД

Вывести на главной странице приложения список логинов всех пользователей, используя ListView и привязку данных

5.5 Создание страницы для работы с записью БД

5.5.1 Добавить в приложение новую страницу UserPage и разместить на ней:

- метки и поля ввода для указания всех данных пользователя кроме Id
- кнопки «Сохранить», «Удалить», «Отмена»

5.5.2 Создать обработчики нажатий на кнопки страницы UserPage, чтобы при сохранении добавлялась/изменялась запись, при удалении – удалялась. При нажатии на каждую из кнопок требуется вернуться на предыдущую страницу.

5.5.3 Добавить на главную страницу кнопку «Добавить» и обработчик нажатия на кнопку и выбора элемента в ListView. В обработчиках реализовать переход к UserPage.

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «SQLite»?

8.2 Какую библиотеку требуется подключить для работы с БД SQLite в приложении xamarin Forms?

8.3 Для чего используется асинхронное подключение к БД?

8.4 Какие операции по работе с данными доступны в БД SQLite?

Практическая работа №27

Взаимодействие с сетью

1 Цель работы

- 1.1 Научиться работать с веб-сервером, используя Xamarin.

2 Литература

2.1 metanit.com : Руководство по программированию для Xamarin Forms : сайт. – 2022. – URL: <https://metanit.com/sharp/xamarin/>. – Режим доступа: свободный. – Текст : электронный. – гл.12,15.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

В качестве веб-сервера использовать <https://jsonplaceholder.typicode.com/>, позволяющий получать и отправлять данные в формате JSON.

5.1 Подключение к сети

Добавить в приложение на главную страницу метку, выводящую:

- информацию о том, есть ли подключение к сети
- список доступных подключений

5.2 Создание классов ViewModel

5.2.1 Добавить в приложение класс Post со следующими открытыми автосвойствами: userId (id пользователя, опубликовавшего пост), id (id поста), title (заголовок поста),

body (текст поста)

5.2.2 Добавить в приложение класс UserViewModel, реализующий интерфейс INotifyPropertyChanged. В класс добавить:

- закрытые поля id, title, body
- метод OnPropertyChanged
- открытые свойства для доступа к полям, при изменении данных в свойствах вызывать OnPropertyChanged

5.3 Получение данных в классе ViewModel

5.3.1 Подключить библиотеку Newtonsoft.Json для работы с JSON

5.3.2 Добавить в класс UserViewModel асинхронный метод LoadData для загрузки данных по сети:

url: <https://jsonplaceholder.typicode.com/posts/1>

5.3.3 Добавить в класс `UserViewModel` открытое автосвойство `LoadDataCommand`

5.3.4 Добавить в класс `UserViewModel` конструктор, присваивающий значение `LoadDataCommand`

5.4 Настройка страницы для отображения данных записи

Указать в главной странице контекст привязки (`BindingContext`) – поле типа `UserViewModel`.

Разместить на странице кнопку и метки для отображения заголовка и текста поста. Привязать кнопку к `LoadDataCommand`, метки – к данным поста (`title`, `body`).

6 Порядок выполнения работы

6.1 Запустить Visual Studio и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «MVVM»?

8.2 Как создать View в приложении Xamarin Forms и для чего он предназначен?

8.3 Как создать ModelView в приложении Xamarin Forms и для чего он предназначен?

8.4 Как выполнить подключение к сети в приложении Xamarin Forms?

8.5 Какое пространство имен требуется подключить для работы с JSON?