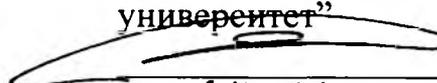


Учреждение образования
«Белорусский государственный экономический университет»

УТВЕРЖДАЮ

Первый проректор учреждения
образования «Белорусский
государственный экономический
университет»


Е.Ф. Киреева
24.02. 2022 г.
Регистрационный № УД 5048.22/уч.

РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности
1-31 03 06 «Экономическая кибернетика» (по направлениям)

Учебная программа составлена на основе типового учебного плана учреждения высшего образования по специальности 1-31 03 06 «Экономическая кибернетика», дата утверждения 30.06.2021, регистрационный номер № G 31-1-028/пр.-тип.

СОСТАВИТЕЛИ:

Тарасов С.А., доцент кафедры математических методов в экономике учреждения образования «Белорусский государственный экономический университет», кандидат технических наук, доцент

РЕЦЕНЗЕНТЫ:

Кашникова Инна Васильевна, заведующий кафедрой микропроцессорных систем и сетей учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат физико-математических наук, доцент;

Зеневич Анна Михайловна, заведующий кафедрой экономической информатики учреждения образования «Белорусский государственный экономический университет», кандидат экономических наук, доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой математических методов в экономике учреждения образования «Белорусский государственный экономический университет»
(протокол № 5 от 12.06.2021)

Научно-методическим советом учреждения образования «Белорусский государственный экономический университет»
(протокол № 5 от 24.06.2021)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

С развитием и внедрением в повседневную жизнь информационно-коммуникационных технологий резко возрос интерес к программированию. Современным специалистам приходится иметь дело с огромным количеством разнообразных языков программирования различных уровней и назначений. Потребность в эффективных средствах разработки программного обеспечения привела к появлению объектно-ориентированных систем программирования, предназначенных для быстрой разработки программных приложений.

Учебная дисциплина "Разработка кросс-платформенных приложений" является общепрофессиональной и направлена на формирование у студента фундамента понимания основ программирования на различных платформах.

Целью изучения учебной дисциплины «Разработка кросс-платформенных приложений» является формирование у обучающихся общепрофессиональных компетенций в процессе изучения кроссплатформенных языков и сред программирования для последующего применения в учебной и практической деятельности.

Задачи преподавания учебной дисциплины:

- приобретение знаний теоретических основ, необходимых для разработки и проектирования кроссплатформенных приложений;
- приобретение знаний о системе классификаций кроссплатформенных приложений;
- формирование и развитие умений использовать полученные знания для создания прикладных программ;
- формирование и развитие умений анализировать производительность получаемых решений;
- приобретение и формирование навыков анализа существующих инструментальных систем, предоставляемых ведущими производителями для создания кроссплатформенных приложений;
- приобретение и формирование навыков работы в интегрированных средах разработки.

В результате изучения учебной дисциплины выпускник должен обладать следующими компетенциями:

– БПК-2. Строить, анализировать и тестировать алгоритмы и программы решения типовых задач обработки информации с использованием структурного, объектно-ориентированного и иных парадигм программирования.

Изучение учебной дисциплины направлено на формирование способности решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

Знать:

- теоретические основы, необходимых для разработки и проектирования кроссплатформенных приложений;
- о системе классификаций кроссплатформенных приложений;

- особенности программирования для различных операционных систем;

Уметь:

- использовать полученные знания для создания прикладных программ;

- анализировать производительность получаемых решений;

- разрабатывать простейшие программы для работы в различных операционных системах;

создавать кроссплатформенные программы на - уровне выполнения

Владеть:

- навыком анализа существующих инструментальных систем, предоставляемых ведущими производителями для создания кроссплатформенных приложений;

- методами отладки и профилирования создаваемых приложений;

- навыками работы с использованием языка программирования Python

В соответствии с учебным планом специальности 1-31 03 06 «Экономическая кибернетика» учебная программа рассчитана на 216 часов, из них аудиторных занятий 102 часов, в том числе: лекций – 52 часов, лабораторных занятий – 50 часов.

Форма текущей аттестации – экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Базовые концепции кросс-платформенного программирования

Понятие кроссплатформенного программного обеспечения. Проблема переносимости приложений с одной ОС на другую. Использование библиотек для создания графического интерфейса пользователя (ГИП или в англоязычной нотации GUI graphical user interface) и системных библиотек конкретной операционной системы (платформозависимых библиотек). Особенности кроссплатформенности java. Использование кроссплатформенных библиотек в стандартном языке (C, C++, Perl, Python, Ruby и др.) Веб-программирование как особый подход в кроссплатформенном программировании. О кроссплатформенных IDE (Integrated development environment).

Тема 2. Особенности программирования для различных операционных систем

Кроссплатформенные среды исполнения. Кроссплатформенные пользовательские интерфейсы. Особенности использования кроссплатформенных библиотек на примере PyQt, PySide. Популярные фреймворки, которые помогут при разработке программ для компьютеров под управлением Windows/MacOS/Linux.

Тема 3. Подготовка к программированию на Python с использованием мобильных устройств

Установка и работа в Pydroid 3 на базе Android. Два основных подхода к работе с интерпретатором Python: непосредственная интерпретация строк кода, вводимых с клавиатуры в интерактивном режиме и выполнение файлов с исходным кодом в пакетном режиме. Вход в интерактивный режим работы. Пакетный режим работы.

Тема 4. Модель данных в языке Python

Python, как язык с неявной сильной динамической типизацией. Отличия динамической и статической типизации, сильной и слабой типизации. Разделение типов данных на встроенные в интерпретатор (*built-in*) и не встроенные, которые можно использовать при импортировании соответствующих модулей. Основные встроенные типы данных. Объявление и инициализация переменных. Объект, как абстракция для представления данных. Идентификатор как некоторое целочисленное значение, посредством которого уникально адресуется объект. Изменяемые и неизменяемые типы данных.

Тема 5. Арифметические операции

Три встроенных числовых типа данных. Арифметические операции с целыми и вещественными числами. Работа с комплексными числами. Доступные битовые операции в Python. Представление чисел в других системах счисления. Библиотека math из стандартной поставки Python.

Тема 6. Работа со строками в Python

Литералы строк. Строки в апострофах и в кавычках. Экранированные последовательности для вставки служебных символов. Использование сырых строк для подавления экранирования. Строки в тройных апострофах и кавычках. Базовые операции для строк: конкатенация, дублирование строки, определение длины строки, доступ по индексу, извлечение среза. Форматирование строк с помощью метода `format`.

Тема 7. Условные операторы и циклы

Конструкция условного оператора ветвления `if`. Использование альтернативного варианта выполнения программы с помощью оператора `if – else`. Реализация выбора из нескольких альтернатив можно с помощью конструкции `if – elif – else`. Оператор цикла `while` для выполнения указанного набора инструкций. Операторы `break` и `continue` для работы с циклами. Оператор `for` для выполнения указанного набора инструкций заданное количество раз.

Тема 8. Работа со списками

Список (`list`) как структура данных для хранения объектов различных типов. Хранение списков в памяти. Создание, изменение, удаление списков и работа с его элементами. Создание копий списков и простое присвоение списков. Добавление и удаление элементов из списка. Методы списков. “Списковое включение” как часть синтаксиса языка для предоставления простого способа построения списков. `List Comprehensions` как обработчик списков. Слайсы (срезы) - составляющая Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. Задание слайса тройкой чисел, разделенных запятой: `start:stop:step` .

Тема 9. Кортежи

Кортеж (`tuple`) как неизменяемая структура данных. Причины, по которым стоит использовать кортежи вместо списков. Создание, удаление кортежей и работа с его элементами. Преобразование кортежа в список и обратно.

Тема 10. Словари

Структура данных, предназначенная для хранения произвольных объектов с доступом по ключу. Хранение данных в формате ключ – значение. Создание, изменение, удаление словарей и работа с его элементами. Проверка наличия ключа в словаре. Методы словарей.

Тема 11. Функции в Python

Функция как именованный фрагмент программного кода, к которому можно обратиться из другого места программы. Использование ключевого слова `def` для создания функции. Возврат значения функцией. Использование функций для обработки данных. Безымянная функция с произвольным числом аргументов.

Тема 12. Работа с исключениями

Понятие исключения в языке программирования. Как исключения дают возможность дальнейшей работы в рамках основного алгоритма. Синхронные и асинхронные исключения. Структурная и неструктурная обработка исключений. Синтаксические ошибки. Иерархия исключений в Python. Обработка исключения внутри синтаксической конструкции `try...except`. Выполнение определенного программного кода при выходе из блока `try/except` с помощью оператора `finally`. Генерация исключений в Python. Пользовательские исключения.

Тема 13. Ввод-вывод данных. Работа с файлами

Вывод данных в консоль. Функция для считывания вводимых с клавиатуры данных. Преобразование строки в список с помощью метода `split()`. Считывание списка чисел с одновременным приведением их к типу `int`. Открытие и закрытие файла. Чтение данных из файла. Запись данных в файл. Дополнительные методы для работы с файлами.

Тема 14. Модули и пакеты. Установка пакетов в Python

Упаковка в модуль классов, объектов, функций и констант. Формирование пространства имен для работы с модулями с помощью пакетов. Как импортировать модули в Python. Задание псевдонима для модуля в разрабатываемой программе. Пакет как каталог, включающий в себя другие каталоги и модули.

Тема 15. Классы и объекты

Три основных “столпа” ООП - инкапсуляция, наследование и полиморфизм. Создание класса в Python с помощью инструкции `class`. Объявление класса, имя класса и тело класса. Создание объекта класса. Статические и динамические атрибуты класса. Методы класса: статические, классовыми (среднее между статическими и обычными) и уровня класса. Конструктор класса и инициализация экземпляра класса. Ключевое слово `self`. Уровни доступа атрибута и метода в Python. Свойство как метод класса, работа с которым подобна работе с атрибутом. Наследование. Родительский класс. Переопределение методов базового класса в классе наследнике на базе полиморфизма.

Тема 16. Итераторы и генераторы

Инструменты, которые, как правило, используются для поточной обработки данных. Использование итератора для упрощения навигации по элементам объекта. Создание собственных итераторов. Упрощение работы по конструированию итераторов с помощью генераторов.

Тема 17. Декораторы функций в Python

Функция как объект специального типа, который можно передавать в качестве аргумента другим функциям. Создание, вызов и возвращение функций из других функций, идея замыкания (`closures`). Декоратор в Python как

некоторая функция, аргументом которой является другая функция. Создание декоратора. Передача аргументов в функцию через декоратор. Декораторы для методов класса. Возврат результата работы функции через декоратор.

Тема 18. Работа с базой данных SQLite

Модуль `sqlite3`, входящий в состав стандартной библиотеки Python и позволяющий работать с базой данных SQLite. Создание новой базы данных. Создание таблицы в базе данных. Добавление записей в таблицу. Обновление и удаление записей. Изменение структуры уже созданной таблицы. Извлечение данных из таблицы. Выбор записей из нескольких таблиц. Ограничение набора выводимых, изменяемых или удаляемых записей с помощью некоторого условия. Индексированные поля. Вложенные запросы. Удаление таблицы и базы данных. Единый интерфейс доступа для работы с базами данных в языке Python. Создание и открытие базы данных. Выполнение запроса к базе данных с помощью методов объекта-курсора. Обработка результата запроса. Создание пользовательской сортировки.

Тема 19. Основы использования библиотеки PyQt5

Библиотека PyQt5, позволяющая разрабатывать приложения с графическим интерфейсом. Установка библиотеки. Структура PyQt-программы. ООП-стиль создания окна. Повторное использование кода при ООП-стиле. Наиболее часто используемые модули PyQt 5.

Тема 20. Управление окном приложения

Создание и отображение окна. Указание типа окна. Изменение и получение размеров окна. Местоположение окна на экране и управление им. Указание координат и размеров прямоугольной области. Разворачивание и сворачивание окна. Изменение цвета фона окна. Программное закрытие окна.

Тема 21. Обработка сигналов и событий в окне приложения

Обработка какого-либо сигнала как необходимость сопоставления ему функции или метода. Методы для блокировки и удаления обработчиков. Генерирование сигнала программно о некотором произошедшем событии. Передача данных в обработчик. Выполнение метода с предопределенным названием через заданный интервал времени с помощью таймера. Перехват всех событий. Методы, предназначенные для обработки определенного события окна. События клавиатуры. События мыши. Работа с буфером обмена. Фильтрация событий.

Тема 22. Размещение компонентов в окнах

Взаимное расположение и минимальные размеры при размещении в окне нескольких компонентов. Абсолютное позиционирование. Горизонтальное и вертикальное выравнивание. Возможность размещения компонентов внутри ячеек сетки. Выравнивание компонентов формы. Состояние одних компонентов при наличии зависимости от состояния других. Добавления рамки вокруг компонента. Панели с изменяемым размером. Область с полосами прокрутки.

Тема 23. Основные компоненты

Надпись для вывода подсказки пользователю, информирования пользователя о ходе выполнения операции. Командная кнопка для запуска какой-либо операции. Переключатели или радиокнопки. Флажок для включения или выключения какой-либо опции. Однострочное текстовое поле для ввода и редактирования текста небольшого объема. Контроль ввода данных. Многострочное текстовое поле. Изменение параметров текста и фона. Текстовый браузер с возможностью перехода по гиперссылкам. Индикатор хода процесса, с помощью которого можно информировать пользователя о текущем состоянии выполнения длительной операции.

Тема 24. Добавление списков и таблиц

Раскрывающийся список с возможностью выбора одного пункта. Добавление, изменение, удаление и получение значений элементов. Изменение параметров списка. Поиск элемента в списке. Элемент списка как хранитель набора величин, каждая из которых относится к определенной роли. Модельное представление для отображения данных в виде списков и таблиц. Доступ к данным внутри модели. Классы представлений для отображения элементов модели. Простой список с возможностью выбора как одного, так и нескольких пунктов. Класс, реализующий таблицу. Иерархический список. Управление заголовками строк и столбцов.

Тема 25. Работа с базами данных

Класс, который отвечает за соединение с базой данных и обработку транзакций. Соединение с базами данных различных форматов. Получение сведений о структуре таблицы. Получение сведений об отдельном поле. Получение сведений об индексе. Выполнение запросов к базе данных. Обработка результатов выполнения запросов. Очистка запроса. Модель, связанная с SQL-запросом. Модель, связанная с таблицей. Модель, поддерживающая межтабличные связи.

Тема 26. Работа с графикой

Все компоненты, которые видимы, на самом деле нарисованы. Вызов события перерисовки компонента. Вспомогательные классы, с помощью которых производится настройка различных параметров: цвета, характеристик шрифта, стиля пера и кисти. Класс, содержащий необходимые средства, позволяющие выполнять рисование геометрических фигур. Вывод текста. Методы для вывода растровых изображений. Сохранение команд рисования в файл. Возможности работы с изображениями.

**Учебно-методическая карта учебной дисциплины «Разработка кросс-платформенных приложений»
для дневной формы получения высшего образования
для специальности
1-31 03 06 «Экономическая кибернетика» (по направлениям)**

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов						Иное*	Форма контроля знаний	
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСП				
						Л	Пз			Лаб
1	Базовые концепции кросс-платформенного программирования	2						[5]	Тест	
2	Особенности программирования для различных операционных систем	2						[5]	Тест	
3	Подготовка к программированию на Python с использованием мобильных устройств	2						[4 д]	Проверка работоспособности устройства и готовности к выполнению им программ	
4	Модель данных в языке Python	2						[1,2]	Тест	
5	Арифметические операции	2						[1,2]	Контрольный запуск заданной программы	
6	Работа со строками в Python	2			2			[1,2]	Тест + контрольный запуск заданной программы	
7	Условные операторы и циклы	2			2			[1, 2]	Тест + контрольный запуск заданной программы	
8	Работа со списками	2			2			[1, 2]	Тест + контрольный запуск заданной программы	
9	Кортежи	2						[1, 2]	Тест	
10	Словари	2			2			[1, 2]	Тест + контрольный запуск заданной программы	
11	Функции в Python	2			2			[3]	Тест + контрольный запуск заданной программы	
12	Работа с исключениями	2			2			[3]	Тест + контрольный запуск заданной программы	
13	Ввод-вывод данных. Работа с файлами	2			2			[1,3]	Тест + контрольный запуск заданной программы	
14	Модули и пакеты. Установка пакетов в Python	2						[1, 3]	Тест	

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов						Иное*	Форма контроля знаний	
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР				
						Л	Пз			Лаб
15	Классы и объекты	2			4				[3, 4]	Тест + контрольный запуск заданной программы
16	Итераторы и генераторы	2			2				[4]	Тест
17	Декораторы функций в Python	2			2				[3, 4]	Тест + контрольный запуск заданной программы
18	Работа с базой данных SQLite	2			4				[1 д]	Тест + контрольный запуск заданной программы
19	Основы использования библиотеки PyQt5	2							[2 д]	Тест
20	Управление окном приложения	2			4				[2 д]	Тест + контрольный запуск заданной программы
21	Обработка сигналов и событий в окне приложения	2			4				[2 д]	Тест + контрольный запуск заданной программы
22	Размещение компонентов в окнах	2			4				[2 д]	Тест + контрольный запуск заданной программы
23	Основные компоненты	2			2				[2 д]	Тест + контрольный запуск заданной программы
24	Добавление списков и таблиц	2			4				[2 д]	Тест + контрольный запуск заданной программы
25	Работа с базами данных	2			4				[2 д]	Тест + контрольный запуск заданной программы
26	Работа с графикой	2			2				[2 д]	Тест + контрольный запуск заданной программы
	Всего часов	52			50					

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Методические рекомендации по организации самостоятельной работы студентов по учебной дисциплине «Разработка кросс-платформенных приложений»

В овладении знаниями учебной дисциплины важным этапом является самостоятельная работа студентов. Рекомендуется бюджет времени для самостоятельной работы в среднем 1,5-2 часа на 2-х часовое аудиторное занятие.

Основными направлениями самостоятельной работы студента являются:

- первоначально подробное ознакомление с программой учебной дисциплины;
- ознакомление со списком рекомендуемой литературы по учебной дисциплине в целом и ее разделам, наличие ее в библиотеке и других доступных источниках, изучение необходимой литературы по теме, подбор дополнительной литературы;
- изучение и расширение лекционного материала преподавателя за счет специальной литературы, консультаций;
- подготовка к лабораторным занятиям по специально разработанным планам с изучением основной и дополнительной литературы;
- подготовка к выполнению диагностических форм контроля (контрольные задания на компьютере);
- подготовка к экзамену.

Примерный перечень тем управляемой самостоятельной работы обучающихся:

УСРС в данном курсе не предусмотрена учебным планом;

Средства диагностики результатов учебной деятельности:

- краткий опрос по синтаксису по материалам прошедших лекций и лабораторным работам перед началом занятий в подгруппах;
- краткий опрос по алгоритмам из материалов прошедших лекций и лабораторным работам перед началом занятий в подгруппах.

ЛИТЕРАТУРА

Основная:

1. Саммерфилд, М. Программирование на Python 3. Подробное руководство [Текст]/ М.Саммерфилд. – СПб.: Символ – Плюс, 2009. – 608 с.
2. Лучано Ромальо. Python. К вершинам мастерства/ Перевод с англ. Слинкин А.А. – М.: ДМК Пресс, 2016 – 768 с.
3. Мэт Харрисон. Как устроен Python. Гид для разработчиков, программистов и интересующихся – СПб.: Питре, 2019. – 272 с.
4. Тони Гэддис Начинаем программировать на Python – СПб.: Bhv, 2019.
5. Матренин П.В. Введение в кроссплатформенное программирование – М.: Нобель -Пресс, 2013. – 55 с.
- 6.Васильев, А. Н. Python на примерах : практический курс по программированию / А. Н. Васильев. - 3-е изд. - Санкт-Петербург : Наука и техника,2019.-428с.:ил.
- 7.Поляков, К. Ю. Программирование. Python. C++ : учебное пособие для общеобразовательных организаций: [В 4 ч.]. / К. Ю. Поляков. - Москва : БИНОМ. Лаборатория знаний, 2019. - Ч. 4. – 192 с. : ил. - (Профильная школа).
- 8.Поляков, К. Ю. Программирование. Python. C++ : учебное пособие для общеобразовательных организаций: [В 4 ч.]. / К. Ю. Поляков. - Москва : БИНОМ. Лаборатория знаний, 2019. - Ч. 2. – 174 с. : ил. - (Профильная школа).
- 9.Поляков, К. Ю. Программирование. Python. C++ : учебное пособие для общеобразовательных организаций: [В 4 ч.]. / К. Ю. Поляков. - Москва : БИНОМ. Лаборатория знаний, 2019. - Ч. 1. – 144 с. : ил. - (Профильная школа).
- 10.Поляков, К. Ю. Программирование. Python. C++ : учебное пособие для общеобразовательных организаций: [В 4 ч.] / К. Ю. Поляков. - 3-е изд., стер. - Москва : Просвещение, 2021. - Ч. 3. – 208 с. : ил. - (Профильная школа).
- 11.Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для студентов высших учебных заведений, обучающихся по инженерно-техническим направлениям / Д. Ю. Федоров. - 2-е изд., перераб. и доп. - Москва : Юрайт, 2021. - 160, [1] с. : ил. - (Высшее образование).
- 12.Языки программирования Python : учебно-методическое пособие. - Минск : БНТУ, 2021. - 90, [1] с.

Дополнительная литература:

1. Пол Бэрри Изучаем программирование на Python, Издательство «Э», 2017. — 624 с.
2. Прохоренок, Н. А. Python 3 и PyQt. Разработка приложений.--: СПб.: БХВ-Петербург, 2012.-704 с.:
3. Манакова, И.П. Информационные системы и технологии. Языки программирования высокого уровня. Программирование на языке Python

[Электронный ресурс]: учеб.-метод, пособие / И.П. Манакова; М-во образования и науки РФ; ФГАОУ ВО «УрФУ им. первого Президента России Б.Н. Ельцина», Нижнетагил. технол. ин-т (филиал). - Нижний Тагил : НТИ (филиал) УрФУ, 2016. – 37 с.

4. Официальный сайт языка Python. URL: <http://python.org/> (дата обращения: 20.12.2021).

5. Силен, Д. Основы Data Science и Big Data. Python и наука о данных / Д. Силен, А. Мейсман, М. Али ; [перевел с англ. Е. Матвеев]. - Санкт-Петербург [и др.] : Питер, 2020. - 334 с. : ил. - (Библиотека программиста).

6. Жуков, Р. А. Язык программирования Python : практикум : учебное пособие для учебных заведений, реализующих программу среднего профессионального образования по укрупненным группам специальностей 09.02.00 "Информатика и вычислительная техника", 38.02.00 "Экономика и управление" / Р. А. Жуков ; Финансовый ун-т при Правительстве Рос. Федерации. - Москва : ИНФРА-М, 2021. - 214, [1] с. : ил. - (Среднее профессиональное образование).

7. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие : [16+] / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500056> (дата обращения: 02.02.2022). – Библиогр. в кн. – ISBN 978-5-9275-2649-9. – Текст : электронный.

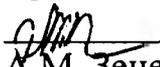
8. Шелудько, В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули : учебное пособие : [16+] / В. М. Шелудько. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2017. – 108 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500060> (дата обращения: 02.02.2022). – Библиогр. в кн. – ISBN 978-5-9275-2648-2. – Текст : электронный.

9. Беспалов, Д. А. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения : учебное пособие: [16+] / Д. А. Беспалов, С. М. Гушанский, Н. М. Коробейникова ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – Часть 1. – 140 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=577698> (дата обращения: 02.02.2022). – Библиогр. в кн. – ISBN 978-5-9275-3367-1. – Текст : электронный.

10. Беспалов, Д. А. Операционные системы реального времени и технологии разработки кроссплатформенного программного обеспечения : учебное пособие : [16+] / Д. А. Беспалов, С. М. Гушанский, Н. М. Коробейникова ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2019. – Часть 2. – 169 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=577699> (дата обращения: 02.02.2022). – Библиогр. в кн. – ISBN 978-5-9275-3368-8. – Текст : электронный.

11. Stemkoski, Lee Developing Graphics Frameworks with Python and OpenGL / Lee Stemkoski, Michael Pascale. – CRC Press, 2022. – 334 p. – DOI: 10.1201/9781003181378

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Компьютерные сети и управление базами данных	Кафедра экономической информатики	Предложений нет  А.М. Зеневич	протокол № <u>5</u> от <u>16.12.2021</u>

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО
на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры
_____ (протокол № ____ от _____ 20_ г.)

Заведующий кафедрой

(подпись)

(И.О.Фамилия)

УТВЕРЖДАЮ
Декан факультета

(подпись)

(И.О.Фамилия)