

Учреждение образования
«Белорусский государственный экономический университет»

УТВЕРЖДАЮ

Ректор учреждения образования
«Белорусский государственный
экономический университет»

_____ А.В. Егоров
_____ 2024 г.

Регистрационный № УД 6436-Муч.

РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности
6-05-0533-09 «Прикладная математика»

Учебная программа составлена на основе образовательного стандарта общего высшего образования ОСВО 6-05-0533-09-2023 и учебного плана по специальности 6-05-0533-09 «Прикладная математика» (профилизация «Экономическая кибернетика»).

СОСТАВИТЕЛИ:

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

РЕЦЕНЗЕНТЫ:

И.В.Кашникова, заведующий кафедрой микропроцессорных систем и сетей обособленного подразделения «Институт информационных технологий» учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат физико-математических наук, доцент;

Л.В.Серебряная, 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 № 4 от 15.11 2024);

Методической комиссией по специальностям «Экономическая кибернетика (по направлениям)», «Прикладная математика», «Экономика» с профилизацией «Анализ данных в экономике и бизнесе» учреждения образования «Белорусский государственный экономический университет» (протокол № 3 от 15.11.2024).

Научно-методическим советом учреждения образования «Белорусский государственный экономический университет» (протокол № 2 от 18.12 2024)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Разработка кросс-платформенных приложений» направлена на формирование у студента фундамента понимания основ программирования на различных платформах.

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.

Цель преподавания учебной дисциплины «Разработка кросс-платформенных приложений» заключается в формировании у обучающихся общепрофессиональных компетенций в процессе изучения кроссплатформенных языков и сред программирования для последующего применения в учебной и практической деятельности.

Достижение поставленной цели предполагает решение следующих **задач**:

- 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 «Разработка кросс-платформенных приложений» формируются следующие компетенции **универсальная**:

- решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий;

базовые профессиональные:

- применять навыки построения, анализа и тестирования алгоритмов и программ для решения типовых задач прикладной математики;

- применять при проектировании приложений в интегрированных средах разработки такие парадигмы программирования как структурное, объектно ориентированное и функциональное программирование.

В результате изучения учебной дисциплины студент должен

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

иметь навык:

- анализа существующих инструментальных систем, предоставляемых ведущими производителями для создания кроссплатформенных приложений;
- отладки и профилирования создаваемых приложений;
- работы с использованием языка программирования Python

В рамках образовательного процесса по данной учебной дисциплине студент должен приобрести не только теоретические и практические знания, умения и навыки по специальности, но и развивать свой ценностно-личностный, духовный потенциал, сформировать качества патриота и гражданина, готового к активному участию в экономической, производственной, социально-культурной и общественной жизни страны.

Учебная дисциплина «Разработка кросс-платформенных приложений» относится к модулю «Программирование» государственного компонента.

Учебные дисциплины, усвоение которых необходимо для изучения данной учебной дисциплины – «Основы и методологии программирования». Учебные дисциплины, для усвоения которых важна данная дисциплина – «Технологии программирования».

Изучение 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.

Форма получения высшего образования - дневная.

В соответствии с учебным планом университета на изучение учебной дисциплины отводится: 216 часов, из них аудиторных занятий 102 часов, в том числе: лекций – 52 часов, лабораторных занятий – 50 часов.

Трудоемкость – 6 з.ед.

Форма промежуточной 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.

Требования к курсовой работе

Курсовой работы учебным планом не предусмотрено.

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
«РАЗРАБОТКА КРОСС-ПЛАТФОРМЕННЫХ ПРИЛОЖЕНИЙ»**

Дневная форма получения высшего образования

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов							Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР				
						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							[1,2]	Тест + контрольный запуск заданной программы
	Работа со строками в Python				2				[1,2]	Тест + контрольный запуск заданной программы
7	Условные операторы и циклы	2							[1, 2]	Тест + контрольный запуск заданной программы
	Условные операторы и циклы				2				[1, 2]	Тест + контрольный запуск заданной программы
8	Работа со списками	2							[1, 2]	Тест + контрольный запуск заданной программы

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов							Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР				
						Л	Пз	Лаб		
	Работа со списками				2				[1, 2]	Тест + контрольный запуск заданной программы
9	Кортежи	2							[1, 2]	Тест
10	Словари	2							[1, 2]	Тест + контрольный запуск заданной программы
	Словари				2				[1, 2]	Тест + контрольный запуск заданной программы
11	Функции в Python	2							[3]	Тест + контрольный запуск заданной программы
	Функции в Python				2				[3]	Тест + контрольный запуск заданной программы
12	Работа с исключениями	2							[3]	Тест + контрольный запуск заданной программы
	Работа с исключениями				2				[3]	Тест + контрольный запуск заданной программы
13	Ввод-вывод данных. Работа с файлами	2							[1,3]	Тест + контрольный запуск заданной программы
	Ввод-вывод данных. Работа с файлами				2				[1,3]	Тест + контрольный запуск заданной программы
14	Модули и пакеты. Установка пакетов в Python	2							[1, 3]	Тест
15	Классы и объекты	2							[3, 4]	Тест + контрольный запуск заданной программы
	Классы и объекты				4				[3, 4]	Тест + контрольный

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов							Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР				
						Л	Пз	Лаб		
										запуск заданной программы
16	Итераторы и генераторы	2							[4]	Тест
	Итераторы и генераторы				2				[4]	Тест
17	Декораторы функций в Python	2							[3, 4]	Тест + контрольный запуск заданной программы
	Декораторы функций в Python				2				[3, 4]	Тест + контрольный запуск заданной программы
18	Работа с базой данных SQLite	2							[1 д]	Тест + контрольный запуск заданной программы
	Работа с базой данных SQLite				4				[1 д]	Тест + контрольный запуск заданной программы
19	Основы использования библиотеки PyQt5	2							[2 д]	Тест
20	Управление окном приложения	2							[2 д]	Тест + контрольный запуск заданной программы
	Управление окном приложения				4				[2 д]	Тест + контрольный запуск заданной программы
21	Обработка сигналов и событий в окне приложения	2							[2 д]	Тест + контрольный запуск заданной программы
	Обработка сигналов и событий в окне приложения				4				[2 д]	Тест + контрольный запуск заданной программы

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов							Литература	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Количество часов УСР				
						Л	Пз	Лаб		
22	Размещение компонентов в окнах	2							[2 д]	Тест + контрольный запуск заданной программы
	Размещение компонентов в окнах				4				[2 д]	Тест + контрольный запуск заданной программы
23	Основные компоненты	2							[2 д]	Тест + контрольный запуск заданной программы
	Основные компоненты				2				[2 д]	Тест + контрольный запуск заданной программы
24	Добавление списков и таблиц	2							[2 д]	Тест + контрольный запуск заданной программы
	Добавление списков и таблиц				4				[2 д]	Тест + контрольный запуск заданной программы
25	Работа с базами данных	2							[2 д]	Тест + контрольный запуск заданной программы
	Работа с базами данных				4				[2 д]	Тест + контрольный запуск заданной программы
26	Работа с графикой	2							[2 д]	Тест + контрольный запуск заданной программы
	Работа с графикой				2				[2 д]	Тест + контрольный запуск заданной программы
	Всего часов	52			50					Экзамен

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Литература

Основная:

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. – М. : ИНФРА-М, 2025. – 343 с. – (Высшее образование). - ISBN 978-5-16-020255-6. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2166199> (дата обращения: 06.12.2024). – Режим доступа: по подписке.
2. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2022. - 242 с. - ISBN 978-5-9275-4108-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/2057604> (дата обращения: 06.12.2024). – Режим доступа: по подписке.
3. Лучано Ромальо. Python. К вершинам мастерства / Перевод с англ. Слинкин А.А. – М.: ДМК Пресс, 2016 – 768 с.
4. Матренин, П. В. Введение в кроссплатформенное программирование / П. В. Матренин. – М.: Нобель -Пресс, 2013. – 55 с.
5. 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. – М. : Юрайт, 2021. – 160, [1] с.

Дополнительная:

1. Бэрри, П. Изучаем программирование на Python / Пол Бэрри ; [перевод с английского: М. А. Райтман]. – [2-е изд.]. – М. : Э, 2017. – 618 с.
2. Воробьев, Г. А. Основы программирования на Python : учебно-методическое пособие / Г. А. Воробьев ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2022. – 89 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=700515> (дата обращения: 06.12.2024). – Библиогр. в кн. – ISBN 978-5-907461-84-0. – Текст : электронный.
3. Гаско, Р. Простой Python просто с нуля / Р. Гаско ; под ред. Н. Ю. Комлева. - М. : СОЛОН-ПРЕСС, 2023. - 256 с. - (Серия «Программирование»). - ISBN 978-5-91359-334-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2185854> (дата обращения: 06.12.2024). – Режим доступа: по подписке.

4. Гэддис, Т. Начинаем программировать на Python : [перевод с английского] / Тони Гэддис. – СПб. : БХВ-Петербург, 2019. – XXIII, 854 с.

5. Дубров, Д. В. Система построения проектов CMake : Учебник / Д. В. Дубров. – Ростов-на-Дону : Южный федеральный университет, 2015. – 419 с. – ISBN 978-5-9275-1852-4. – URL:

https://www.elibrary.ru/download/elibrary_27266282_11113412.pdf (дата обращения: 06.12.2024).

6. Калитвин, В. А. Введение в программирование на Python : учебное пособие / В. А. Калитвин ; Липецкий государственный педагогический университет им. П. П. Семенова-Тян-Шанского. – Липецк : Липецкий государственный педагогический университет им. П.П. Семенова-Тян-Шанского, 2023. – 85 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=714538> (дата обращения: 06.12.2024). – Библиогр. в кн. – ISBN 978-5-907655-86-7. – Текст : электронный.

7. Карякин, М. И. Технологии программирования и компьютерный практикум на языке Python : учебное пособие / М. И. Карякин, К. А. Ватульян, Р. М. Мнухин ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2022. – 244 с. : ил., табл. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=698687> (дата обращения: 06.12.2024). – Библиогр. в кн. – ISBN 978-5-9275-4108-9. – Текст : электронный.

8. Кондратьева, В. А. Основы программирования на языке Python : учебное пособие в 2-х частях / В. А. Кондратьева. Том Часть 1. – Москва : Московский городской педагогический университет, 2021. – 92 с. – URL: https://www.elibrary.ru/download/elibrary_46438002_62545907.pdf (дата обращения: 06.12.2024).

9. 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.

10. Прохоренок, Н. А. Python 3 и PyQt. Разработка приложений / Николай Прохоренок. – СПб. : БХВ-Петербург, 2012. – 703 с.

11. Саммерфилд, М. Программирование на Python 3 : подробное руководство : [пер. с англ.] / Марк Саммерфилд. – СПб. ; М. : Символ, 2009. – 607 с.

9. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. - 146 с. - ISBN 978-5-9275-2649-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1021662> (дата обращения: 06.12.2024). – Режим доступа: по подписке.

10. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-наДону ; Таганрог : Издательство Южного федерального университета, 2017. - 107 с. - ISBN 978-5-9275-2648-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1021664> (дата обращения: 06.12.2024). – Режим доступа: по подписке.

11. Языки программирования. Python : учебно-методическое пособие для студентов специальности 1-40 01 01 "Программное обеспечение информационных технологий". В 2 ч. Ч. 1 / [В.В. Иванченко и др.] ; М-во образования Респ. Беларусь, Белорус. нац. техн. ун-т, Каф. "Програм. обеспечение информ. систем и технологий". – Минск : БНТУ, 2021. – 90, [1] с.

Интернет-ресурсы:

1. Официальный сайт языка Python. – URL: <http://python.org/> (дата обращения: 06.12.2024).

Перечень вопросов для проведения экзамена

1. Понятие кроссплатформенного программного обеспечения. Проблема переносимости приложений с одной ОС на другую.
2. Особенности кроссплатформенности java. Использование кроссплатформенных библиотек в языках C++ и Python.
3. Кроссплатформенные IDE, кроссплатформенные среды исполнения, кроссплатформенные пользовательские интерфейсы.
4. Типизация языка Python.
5. Типы данных в Python.
6. Создание объектов в памяти, процесс объявления новых переменных.
7. Изменяемые и неизменяемые типы данных.
8. Расширенный набор математических операций. Работа с комплексными числами.
9. Битовые операции. Представление чисел в других системах счисления.
10. Строка как последовательность символов. Экранирование.
11. Базовые операции со строками.
12. Оператор ветвления if и его разновидности.
13. Оператор цикла while. Операторы break и continue.
14. Оператор цикла for.
15. Список (list) как структура данных.
16. Использование спискового включения (*list comprehensions*).
17. Слайсы (срезы) как мощная составляющая Python.
18. Списковая арифметика.
19. Кортеж (tuple) в качестве неизменяемой структуры данных.
20. Преобразование кортежа в список и обратно. Распаковка кортежа.
21. Встроенная структура данных в Python – словарь: варианты создания, добавление и удаление элемента, доступ к элементу словаря.
22. Создание словаря из последовательностей.
23. Допустимые типы ключей словаря.
24. Шаблон для построения функции.
25. Использование функции для обработки данных.
26. Что может возвращать функция. Как вернуть функцию из функции.
27. Аргументы функции.
28. Анонимные функции.
29. Опции, доступные для передачи аргументов в лямбда-выражения.
30. Синхронные и асинхронные исключения. Механизм обработки исключений. Синтаксические ошибки.

31. Примеры исключений в Python. Обработка исключений в Python.
32. Пользовательские исключения (User-defined Exceptions) в Python.
33. Вывод данных в консоль.
34. Ввод данных с клавиатуры, ввод с преобразованием.
35. Создание объекта файл с помощью функции `open()`.
36. Необязательный параметр `mode` в функции `open()`.
37. Запись данных в файл.
38. Параметры в `open()` с третьего по шестой.
39. Чтение данных из файла с помощью методов `read(размер)` и `readline()`.
40. Дополнительные методы для работы с файлами `tell()` и `seek(позиция)`, применение оператора `with`.
41. Разделение на модули и программы. Использование псевдонима для модуля. Импорт через конструкцию `from ... import...` Импортирование нескольких функций из модуля.
42. Повторная загрузка модулей.
43. Пакет в Python. Пакеты как средства распределения модулей по папкам.
44. Введение понятия объекта от концепции черного ящика.
45. Понятия класса и объекта, поля и методы.
46. Три основных “столпа” ООП - это инкапсуляция, наследование и полиморфизм.
47. Создание классов и объектов. Статические и динамические атрибуты класса.
48. Методы статическими, классовыми и уровня класса.
49. Конструктор класса и инициализация экземпляра класса. Ключевое слово `self`.
50. Уровни доступа атрибута и метода в Python.
51. Свойство как метод класса.
52. Наследование в Python.
53. Полиморфизм с позиции переопределения методов базового класса в классе наследнике.
54. Множественное наследование в Python.
55. Расширения функциональности классов с помощью так называемых примесей.
56. Абстрактный класс и его некоторые особенности.
57. Основное назначение итераторов. Использование функции `iter()`, а для извлечения следующего элемента из итератора – функции `next()`.
58. Построение собственного итератора.
59. Контейнеры-последовательности, аналогичные спискам и кортежам.

60. Перечисление как определенный программистом набор каких-либо именованных значений.
61. Декоратор в Python как некоторая функция, аргументом которой является другая функция.
62. Передача аргументов в функцию через декоратор.
63. Возврат результата работы функции через декоратор.
64. Программа, выдающая простое окно с использованием PyQt5.
65. ООП-стиль создания окна с использованием PyQt5.

Перечень лабораторных занятий

1. Работа со строками в Python
2. Условные операторы и циклы
3. Работа со списками
4. Словари
5. Функции в Python
6. Работа с исключениями
7. Ввод-вывод данных. Работа с файлами
8. Классы и объекты
9. Итераторы и генераторы
10. Декораторы функций в Python
11. Работа с базой данных SQLite
12. Управление окном приложения PyQt5
13. Обработка сигналов и событий в окне приложения
14. Размещение компонентов в окнах
15. Основные компоненты PyQt5
16. Добавление списков и таблиц
17. Работа с базами данных
18. Работа с графикой PyQt5

Перечень компьютерных программ, наглядных и других пособий, методических указаний и материалов, технических средств обучения, оборудования для выполнения лабораторных работ

1. Программная среда Python 3
2. База данных SQLite
3. Кроссплатформенная библиотека PyQt5

Организация самостоятельной работы студентов

Для получения компетенций по учебной дисциплине важным этапом является самостоятельная работа студентов.

На самостоятельную работу обучающегося дневной формы получения образования отводится 114 часов.

Содержание самостоятельной работы обучающихся включает все темы учебной дисциплины из раздела «Содержание учебного материала».

При изучении учебной дисциплины используются следующие формы самостоятельной работы:

- самостоятельная работа в виде решения индивидуальных задач в аудитории во время проведения лабораторных занятий под контролем преподавателя в соответствии с расписанием учебных занятий;
- углубленное изучение разделов, тем, отдельных вопросов, понятий;
- выполнение контрольных запусков заданных программ;
- выполнение тестов;
- подготовку к сдаче промежуточной аттестации.

Контроль качества усвоения знаний

Диагностика качества усвоения знаний проводится в рамках текущей и промежуточной аттестаций.

Мероприятия *текущей аттестации* проводятся в течение семестра и включают в себя следующие формы контроля:

выполнение контрольных запусков заданных программ;

тест;

Текущая аттестация по учебной дисциплине проводится три раза в семестр.

Результаты текущей аттестации за семестр, полученные в ходе проведения мероприятий текущей аттестации, оцениваются отметкой в баллах по десятибалльной шкале и отражаются в ведомости текущей аттестации по учебной дисциплине.

Требования к обучающемуся при прохождении промежуточной аттестации.

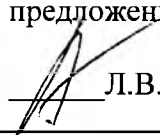
Обучающиеся допускаются к промежуточной аттестации по учебной дисциплине при условии успешного прохождения текущей аттестации (выполнения мероприятий текущего контроля) по учебной дисциплине, предусмотренной в текущем семестре данной учебной программой.

Промежуточная аттестация проводится в форме экзамена.

Методика формирования отметки по учебной дисциплине

В соответствии с Положением о рейтинговой системе оценки знаний, умений и навыков студентов БГЭУ.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ С ДРУГИМИ УЧЕБНЫМИ ДИСЦИПЛИНАМИ

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)
Основы и методологии программирования	Кафедра экономической информатики	Замечаний и предложений нет  Л.В. Серебряная	<i>Протокол №</i> <i>от 15.11.24</i>

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
«МЕТОДЫ ОПТИМИЗАЦИИ»

(Регистрационный № _____ от _____)
на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры математических методов в экономике
(протокол № ____ от _____ 20__ г.)

Заведующий кафедрой

УТВЕРЖДАЮ
Декан факультета
