

Учреждение образования
«Белорусский государственный экономический университет»

УТВЕРЖДАЮ

Первый проректор учреждения
образования «Белорусский
государственный экономический
университет»

 Е.Ф.Киреева

26.06

2023 г.

Регистрационный № УД 5935-23/уч.

Основы и методологии программирования
Учебная программа учреждения высшего образования
по учебной дисциплине для специальности
6-05-0533-09 Прикладная математика

Учебная программа составлена на основе примерного учебного плана учреждения высшего образования по специальности 6-05-0533-09 «Прикладная математика» (профилизация «Вероятность, статистика и анализ данных»), дата утверждения 20.12.2022, регистрационный номер № 6-05-05-020/пр.

СОСТАВИТЕЛИ:

Кравчук А.С., профессор кафедры экономической информатики учреждения образования «Белорусский государственный экономический университет», доктор физ.-мат. наук, доцент

РЕЦЕНЗЕНТЫ:

Кремень Ю.А., доцент кафедры веб-технологий и компьютерного моделирования учреждения образования «Белорусский государственный университет», кандидат физико-математических наук, доцент;

Говядинова Н.Н., доцент кафедры информационных технологий учреждения образования «Белорусский государственный экономический университет», доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой экономической информатики учреждения образования «Белорусский государственный экономический университет» (протокол № 12 от 21.04.2023)

Научно-методическим советом учреждения образования «Белорусский государственный экономический университет» (протокол № 7 от 21.06.2023)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Стремительное развитие информационных технологий и их повсеместное применение, а также повышение доли компьютерных технологий в современной экономике привело к увеличению спроса на специалистов, имеющих подготовку, связанную с навыками обработки информации. Изменение внешних рыночных условий, как следствие, приводит к необходимости переориентации подготовки учащихся на всех уровнях образования в контексте парадигмального изменения и смены акцентуации на более углубленную подготовку в области программирования.

Основной тенденцией развития современной экономики является повышение количества компьютеризированных средств, применяемых во всех отраслях человеческой деятельности. Вследствие этого навыки работы с подобными системами являются необходимой компетенцией для большинства современных профессий.

Поэтому необходимо более детально проработать вопрос изучения информационных технологий в рамках всех ступеней образования и в отдельности рассмотреть компетенцию программирования, так как именно обучение данной практической деятельности позволяет наиболее эффективно развивать универсальные учебные действия, связанные с такими областями, как решение задач, анализ, синтез, декомпозиция, абстрагирование, дедукция и специфические мыслительные навыки, связанные с логическим, критическим и системным мышлением.

Цель преподавания учебной дисциплины «Основы и методологии программирования» – формирование у студентов базовых знаний, умений и навыков, направленных на разработку эффективных прикладных программ в организациях.

Задача учебной дисциплины «Основы и методологии программирования» – не только сформировать у студентов навыки процедурного программирования достаточные для выполнения расчетов, связанных с решением комплекса экономических задач любой сложности, но и создать надежную базу для применения студентами парадигмы объектно-ориентированного при создании программных продуктов.

В результате изучения учебной дисциплины выпускник должен обладать следующими компетенциями:

- УК-2. Решать стандартные задачи профессиональной деятельности на основе применения информационно-коммуникационных технологий;
- БПК-4. Применять навыки построения, анализа и тестирования

алгоритмов и программ для решения типовых задач прикладной математики;

– БПК-5. Применять при проектировании приложений такие парадигмы программирования как структурное, объектно-ориентированное и функциональное программирование, а также иные парадигмы, разрабатывать программное обеспечение в интегрированных средах разработки.

В результате изучения учебной дисциплины студент должен:

знать:

– основные понятия: алгоритмы, базовые конструкции для записи алгоритмов, циклические алгоритмы; простейшие типы данных: целый, вещественный, символьный, массивы данных;

– современный язык процедурного программирования (C/C++);

– способы организации ввода и вывода информации;

– простейшие алгоритмы обработки данных: вычисления по формулам, последовательный поиск, методы сортировки, итерационные алгоритмы поиска корней уравнений, простейшие вычисления определенных интегралов, рекурсивные и итерационные алгоритмы обработки данных, идеи реализации рекурсивных вызовов в функциях;

– структуры данных: одномерный массив, двумерный массив, структура, список, хранение структур данных в памяти компьютера;

– принципы процедурного и объектно-ориентированного программирования.

уметь:

– реализовать алгоритмы, возникающие в задачах обработки данных в области экономики;

– использовать динамические структуры данных при реализации прикладных алгоритмов

– разрабатывать программы для решения практических задач организаций;

владеть:

– навыками планирования разработки программ;

– навыками написания программ.

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 «Основы и методологии программирования» входит в государственный компонент, относится к модулю «Программирование» и изучается в тесной связи с учебными дисциплинами

«Машинно-ориентированное программирование», «Промышленное программирование» и «Разработка кросс-платформенных приложений».

В соответствии с примерным учебным планом специальности 6-05-0533-09 Прикладная математика (профилизация Вероятность, статистика и анализ данных) учебная программа рассчитана на 216 часов, из них аудиторных занятий 132 часов, в том числе: лекций – 64 часов, из них 18 часов УСРС, лабораторных занятий – 68 часов, из них 20 часов УСРС.

Форма получения высшего образования - очная.

Форма текущей аттестации – зачет и экзамен.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Тема 1. Позиционные системы счисления, используемые в современной вычислительной технике. Хранение информации в памяти машины

Основные понятия теории позиционных систем счисления, правила перевода из одной системы счисления в другую. Примеры применения этих правил. Правила хранения информации в памяти компьютера на примере использования 16-битной операционной системы.

Тема 2. Основы алгоритмизации

Изобразительные средства для описания (представление) алгоритма. Алгоритмические языки. Операторные схемы алгоритмов. Псевдокод. Типы алгоритмических процессов. История языков программирования. Стандарты языка C++. Проектирование программного обеспечения. Основные этапы разработки программ

Тема 3. Введение в язык C++

Алфавит. Идентификаторы. Служебные слова. Лексемы. Структура программы. Начальные сведения о директивах препроцессора. Комментарии. Общий вид программы, оформленной в одном файле. Типы данных. Переменные. Простейшие средства ввода-вывода. Код простейшей программы. Общие сведения о константах. Неименованные константы. Именованные константы. Простейший вид макроподстановки. Операции и выражения с использованием переменных и констант базовых типов. Приведение типов. Определение оператора. Пустой оператор. Требования к синтаксису операций и операторов. Хранение и обработка двоичного кода. Системы счисления. Перевод из десятичной в двоичную систему счисления и обратно. Основы представления информации в памяти компьютера. Стандартные математические функции. Средства форматирования ввода/вывода. Использование средств форматированного вывода языка C.

Тема 4. Операторы управления программой

Составной оператор. Область видимости переменных. Условные операторы. Требования к оформлению условных операторов. Операторы циклов. Схема бесконечного цикла. Вложенные циклы. Требования к оформлению операторов циклов. Операторы перехода. Организация вычислений с точностью. Суммирование отрезка степенного ряда с точностью. Точность вычислений согласно рекуррентным уравнениям.

Тема 5. Автоматические массивы

Одномерные автоматические массивы. Примеры простейших действий над одномерными автоматическими массивами. Операции линейной алгебры для векторов в контексте их применения к одномерным массивам. Использование в программе меньшего количества элементов массива, чем задано при объявлении. Строки. Многомерные автоматические массивы. Примеры элементарных действий с двумерным массивом. Заполнение двумерного массива по шаблону. Транспонирование квадратного двумерного массива. Ввод с клавиатуры «усеченного» двумерного массива. Заполнение двумерного массива псевдослучайными числами.

Тема 6. Введение в функции. Стек

Объявление функций (прототип функции). Определение функции. Вызов функции. Возвращение результата в точку вызова функции. Некоторые примеры функций, возвращающих значение. Возврат значений функцией `main()`. Тип возврата `void`. Игнорирование значений, возвращаемых функциями. Пример оформления функции без прототипа. Допустимые варианты оформления пользовательских функций. Последовательность простейших действий по созданию пользовательской функции, выделением алгоритмической части из главной функции. Примеры простейших функций. Инициализация значений формальных параметров по умолчанию. Использование квалификатора `const` в параметрах функций. Требования к синтаксису функций, создаваемых программистом. Элементы ООП в императивном программировании. Перегрузка функций. Параметры функции `main()`. Стек, как структура данных. Стек вызовов функций. Стек на практике. Переполнение стека. Рекурсия.

Тема 7. Создание исполняемого кода. Директивы препроцессора

Трансляция программы: компиляция и интерпретация. Этапы создания исполняемого кода программ на языке C++. Средства управления препроцессорной обработкой программы. Подключаемые файлы. Директива `#include`. Директива `#define`. Директивы условной компиляции. К вопросу об области видимости директивы `#define`. Проблема дублирования объявлений.

Тема 8. Указатели и ссылки

Указатели. Особый тип указателя `void`. Нулевые указатели. Указатели на константы. Константный указатель. Константный

указатель на константу. Размер указателей. Операции над указателями. Адрес указателя. Указатель на указатель. Указатели и автоматические массивы. Взаимосвязь указателей и одномерных автоматических массивов. Одномерные автоматические массивы указателей. Указатель-на-указатель и двумерные автоматические массивы. Куча. Средства работы с кучей. Работа со скалярными переменными в куче. Взаимодействие стека и кучи при работе со скалярной переменной в куче. Одномерные массивы в куче. Двумерные массивы в куче. Ссылки. Основные сведения о ссылке. Ссылки на неконстантные значения. Ссылки vs указатели. Ссылки на константные значения. Константные ссылки. Ссылки на указатели.

Тема 9. Указатели и ссылки в параметрах функций

Ссылки и указатели на переменные в параметрах функций. Указатели в параметрах функции. Ссылки в качестве параметров функций. Ссылка и указатель в качестве возвращаемого функцией значения. Ссылки и указатели на константы в параметрах функций. Указатели на константы в параметрах функций. Ссылки на константы в параметрах функций. Обработка в функциях массивов с помощью указателей. Функции и одномерные массивы. Обработка двумерных массивов в функциях. Передача в функцию автоматических массивов по ссылке. Использование ссылки на одномерный автоматический массив. Ссылка на двумерный автоматический массив. Использование константных ссылок на массив. Использование в параметрах функций ссылки на указатель.

Тема 10. Указатели и ссылки на функции

Синтаксис создания указателя на функцию. Выбор пользователем функции, решающей задачу. Передача указателя на функцию другой функции через параметры. Выбор пользователем направления сортировки. Массив указателей на функцию. Создание псевдонимов типов указателей на функцию. Ключевое слово `typedef`. Инструкция `using`. Ссылки на функцию.

Тема 11. Макросы проверки ошибок

Инструкция `assert()`. Макрос `NDEBUG`. Инструкция `static_assert()`

Тема 12. Простейшие численные методы

Правило округления чисел. Элементы теории погрешностей. Виды погрешностей. Погрешность арифметических действий над приближенными числами. Элементарные методы решения нелинейных уравнений. Отделение корней уравнения. Метод деления отрезка

пополам (метод дихотомии). Метод простых итераций. Метод Ньютона (метод касательных). Вычисление определенных интегралов. вадратурные формулы левых, правых и средних прямоугольников. Формула трапеций. Нижняя и верхняя суммы Дарбу. Пример вычисления нижней суммы Дарбу. Правило Рунге (правило двойного пересчета). Пример программной реализации вычисления определенного интеграла с заданной точностью. Особенности вычисления определенного интеграла с точностью с помощью сумм Дарбу. Численное решение обыкновенного дифференциального уравнения первого порядка. Описание метода Эйлера. Пример программной реализации метода Эйлера.

Тема 13. Основы ООП. Объединения. Наследование. Списки

Введение в классы. Объявление класса. Объявление объекта класса, указателя на объект и ссылки на объект класса. Выделение памяти под хранение массива объектов в стеке. Доступ к компонентам класса. Операция уточнения имени. Операция косвенной адресации. Выделение памяти под объекты и массивы объектов в куче. Правила оформления имен классов, объектов, полей и методов. Инкапсуляция. Спецификаторы доступа. Особенности работы со структурами. Операция уточнение области видимости имени. Основы построения диаграмм классов. Графическое отображение классов. Отображение отношений между классами. Дружественные функции. Массивы автоматической памяти как свойства классов. Массивы хранящиеся в куче как свойства классов. Указатель `this`. Конструктор копирования. Конструктор как средство преобразования типов. Функции-операции. Перегрузка операций. Правила перегрузки операций. Преобразование типов с помощью функции-операции как метода класса. Списки объектов. Общие представление о наследовании. Создание дочернего класса. Графическое отображение наследования в диаграммах классов. Права доступа при наследовании. Управление спецификаторами доступа при наследовании. Переопределение компонентов родительского класса. Конструкторы и деструкторы при наследовании. Конструкторы с параметрами при наследовании. Наследование и указатели на объекты. Раннее связывание. Виртуальные методы. Позднее связывание. Виртуальный деструктор. Абстрактные классы. Обобщенное понятие полиморфизма. Множественное наследование. Виртуальные классы. Списки из объектов различных классов. Классы `enum`. Объединения. Битовые поля.

Тема 14. Шаблоны

Шаблоны функций. Перегрузка шаблона функции. Удобство применения шаблонов функций. Недостатки шаблонов функций. Шаблоны классов. Использование в шаблонах типов «по умолчанию». Пример класса с двумя обобщенными типами данных. Использование в обобщенных классах аргументов, не являющихся типами (параметр по-пуре). Организация шаблона для хранения односвязного списка. Пример шаблона матрица. Преимущества использования шаблонов классов.

Тема 15. STL. Строки в C++. Контейнеры

Класс complex. Структура pair. Структура tuple. Класс string. Преимущества и недостатки использования класса string в сравнении с встроенным типом char*. Контейнеры. Объявление и использование итераторов. Цикл foreach(). Контейнер vector. Контейнер stack. Контейнер queue. Контейнер deque. Контейнер list (двусвязный список). Контейнер set (множество). Алгоритмы стандартной библиотеки. Функторы. Предикаты. Предикатные классы. Лямбда-функции. Знакомство с синтаксисом. Функциональные адаптеры C++. Ознакомление и описание принципа.

Тема 16. Исключительные ситуации

Использование исключений. Блоки отслеживания и обработки исключительных ситуаций. Перехват исключений.

Тема 17. Работа с файлами средствами C/C++

Ввод и вывод. Стандартные функции для работы с файлами (потоками). Открытие/ закрытие файла. Чтение и запись данных.

Требования к курсовой работе

Курсовой работы не предусмотрено.

**УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ
«ОСНОВЫ И МЕТОДОЛОГИИ ПРОГРАММИРОВАНИЯ»
ДЛЯ ДНЕВНОЙ ФОРМЫ ПОЛУЧЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ**

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
1	Позиционные системы счисления, используемые в современной вычислительной технике. Хранение информации в памяти машины						8		Тест	
2	Основы алгоритмизации						4	[1-5]	Тест + эссе, содержащее блок-схему простейшего алгоритма	
3	Введение в язык C++	6					2	[1-5]	Тест + контрольный запуск заданной программы	
4	Операторы управления программой	4					6	[1-5]	Тест + контрольный запуск заданной программы	
5	Автоматические массивы	4			4			[1-5]	Тест + контрольный запуск заданной программы	
6	Введение в функции. Стек	2			4			[1-5]	Тест + контрольный запуск заданной программы	

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		
7	Создание исполняемого кода. Директивы препроцессора	2							[1, 2]	Тест + контрольный запуск заданной программы
8	Указатели и ссылки	4			6				[1, 2]	Тест + контрольный запуск заданной программы
9	Указатели и ссылки в параметрах функций	6			6				[1, 2]	Тест + контрольный запуск заданной программы
10	Указатели и ссылки на функции	2			2				[1, 2]	Тест + контрольный запуск заданной программы
11	Макросы проверки ошибок	1							[1, 2]	Тест + контрольный запуск заданной программы
12	Простейшие численные методы	3			6				[1, 2]	Тест + контрольный запуск заданной программы
13	Основы ООП. Объединения. Наследование. Списки	12			10				[1, 2]	Тест + контрольный запуск заданной программы
14	Шаблоны				2	4			[1, 2]	Тест + контрольный запуск заданной программы

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	STL. Строки в C++. Контейнеры				4	10				
16	Исключительные ситуации				2	2				
17	Работа с файлами средствами C/C++				2	2				
	Всего часов	46			48	18		20		

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ЛИТЕРАТУРА

Основная:

1. Павловская, Т. А. С/С++. Программирование на языке высокого уровня: для магистров и бакалавров: учебник для студентов высших учебных заведений, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Т. А. Павловская. - Санкт-Петербург : Питер, 2020. – 460 с.
2. Лафоре, Р. Объектно-ориентированное программирование в С++ / Р. Лафоре ; [перевели с англ.: А. Кузнецов, М. Назаров, В. Шрага]. - 4-е изд. - Спб. [и др.] : Питер, 2019. - 923 с.
3. Белодед, Н. Н. Алгоритмизация и программирование : пособие для студентов учреждений высшего образования, осваивающих образовательную программу высшего образования 1-й степени по специальности 1-26 03 01 Управление информационными ресурсами : в 4 ч. / Н. И. Белодед, И. А. Юрча, Ж. И. Щербович ; Академия управления при Президенте Республики Беларусь. – Минск : Академия управления при Президенте Республики Беларусь, 2021–. – Ч. 1: Введение в С/С++. Основные элементы языка. – 2021. – 182, [1] с.
4. Володько, Л. П. Основы алгоритмизации и программирования : учебно-методическое пособие для студентов экономических и технических специальностей различных форм обучения и слушателей факультета дополнительного образования : в 2 ч. / Л. П. Володько. – Пинск : ПолесГУ, 2023. – Ч. 1. – 151 с.

Дополнительная литература:

5. Шилдт Г. Справочник программиста по С/С++ / Г. Шилд – Вильямс, 2019. – 432 с.
6. Чукич И. Функциональное программирование на С++ / И. Чукич – ДМК, 2020. – 360 с.
7. Литвиненко Н. Технология программирования на С++. Начальный курс / Н. Литвиненко – ВHV, 2019. – 288 с.
8. Пацей, Н. В. Основы алгоритмизации и программирования / Н.В. Пацей – Минск: БГТУ, 2010. – 289 с.
9. Васильев, А. Программирование на С++ в примерах и задачах / А.Васильев. – М.: Эксмо, 2018. – 368 с
10. Ашарина, И. В. Объектно-ориентированное программирование в С++: лекции и упражнения: учебное пособие для студентов высших учебных заведений, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника и по направлению подготовки бакалавров "Информатика и вычислительная техника" / И. В. Ашарина. - 2-е изд., перераб. и доп. - М. : Горячая линия - Телеком , 2015. – 335 с.

11. Бансила, М. Решение задач на современном C++. Станьте опытным программистом, решая практические задачи / М. Бансила ; [пер. с англ. А.Н. Киселева]. - М.: ДМК Пресс, 2019. - 301 с.

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 зачету, экзамену.

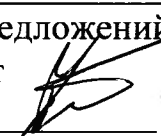
Примерный перечень тем управляемой самостоятельной работы обучающихся:

- Позиционные системы счисления, используемые в современной вычислительной технике. Хранение информации в памяти машины.
- Основы алгоритмизации.
- Введение в язык C++.
- Операторы управления программой.
- Шаблоны.
- STL. Строки в C++. Контейнеры.
- Исключительные ситуации.
- Работа с файлами средствами C/C++.

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.

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)
Машинно-ориентированное программирование	Математических методов в экономике	Предложений нет  Г.О. Читая	Протокол № 12 от «21» апреля 2023 г.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ УВО

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры экономической информатики (протокол № ____ от _____ 202_ г.)

Заведующий кафедрой

к.э.н.

_____ А.М. Зеневич
(подпись) (И.О.Фамилия)

УТВЕРЖДАЮ

Декан факультета

к.э.н., доцент

_____ Т.Н. Налецкая
(подпись) (И.О.Фамилия)