

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учебно-методическое объединение по педагогическому образованию

УТВЕРЖДАЮ

Первый заместитель Министра
образования Республики Беларусь
_____ А.Г.Баханович

Регистрационный № _____

СИСТЕМЫ И ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

**Примерная учебная программа по учебной дисциплине
для специальности**

6-05-0113-04 Физико-математическое образование
(математика и информатика; физика и информатика; информатика)

СОГЛАСОВАНО

Председатель учебно-методического
объединения по педагогическому
образованию

_____ А.И.Жук

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.Н.Пищов

СОГЛАСОВАНО

Начальник Главного управления
общего среднего и дошкольного
образования Министерства
образования Республики Беларусь

_____ М.С.Киндиренко

СОГЛАСОВАНО

Проректор по научно-методической
работе Государственного учреждения
образования «Республиканский
институт высшей школы»

_____ И.В.Титович

Эксперт-нормоконтролер

СОСТАВИТЕЛИ:

Г.А.Заборовский, доцент кафедры информатики и методики преподавания информатики учреждения образования «Белорусский государственный педагогический университет имени Максима Танка», кандидат физико-математических наук, доцент

П.А.Хорошевич, старший преподаватель кафедры информатики и методики преподавания информатики физико-математического факультета учреждения образования «Белорусский государственный педагогический университет имени Максима Танка»

РЕЦЕНЗЕНТЫ:

Кафедра дискретной математики и алгоритмики факультета прикладной математики и информатики Белорусского государственного университета (протокол № 13 от 08.04.2024);

В.В.Сидорик, доцент кафедры программного обеспечения информационных систем и технологий Белорусского национального технического университета, кандидат физико-математических наук, доцент

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ПРИМЕРНОЙ:

Кафедрой информатики и методики преподавания информатики физико-математического факультета учреждения образования «Белорусский государственный педагогический университет имени Максима Танка» (протокол № 8 от 29.03.2024);

Научно-методическим советом учреждения образования «Белорусский государственный педагогический университет имени Максима Танка» (протокол № 6 от 16.04.2024);

Научно-методическим советом по физико-математическому и технологическому образованию учебно-методического объединения по педагогическому образованию (протокол № 1 от 03.04.2024)

Ответственный за редакцию: С.И.Чубаров

Ответственный за выпуск: С.И.Чубаров

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Примерная учебная программа по учебной дисциплине «Системы и технологии программирования» разработана для учреждений высшего образования в соответствии с требованиями образовательного стандарта общего высшего образования по специальности 6-05-0113-04 «Физико-математическое образование (физика и информатика; математика и информатика; информатика)».

Содержание учебной дисциплины «Системы и технологии программирования» предполагает использование в работе со студентами современных образовательных технологий и подходов к разработке и реализации алгоритмов обработки различных структур данных и решения практико-ориентированных заданий из разных предметных областей. Обучение по данной учебной программе направлено на приобретение студентами базовых знаний в области современных систем и технологий программирования, формирование умений создавать творческие проекты. Изучение учебной дисциплины «Системы и технологии программирования» ориентировано на развитие алгоритмического объектно-ориентированного стиля мышления обучающихся, практическую работу с обучающей и развивающей информацией, позволяет стимулировать познавательные интересы, способствует развитию логического и ассоциативного мышления, а также пространственного воображения обучающихся.

Целью учебной дисциплины является формирование профессиональной компетентности учителя информатики в области систем и технологий программирования, обучение созданию компьютерных программ на основе современных систем и технологий программирования.

Задачи учебной дисциплины:

- ознакомление студентов с современными системами, парадигмами и технологиями программирования;
- изучение принципов структурно-модульного, объектно-ориентированного, событийного и визуального программирования;
- демонстрация возможностей выбора технологий, инструментов и методов решения практических задач и анализа полученных решений;
- формирование навыков разработки, тестирования, отладки и модификации программ с использованием современных систем программирования.

Учебная дисциплина «Системы и технологии программирования» базируется на знаниях, полученных при изучении дисциплин «Методы алгоритмизации и программирование», «Программирование в визуализированных средах». Результаты изучения учебной дисциплины способствуют успешному овладению учебных дисциплин модулей «Методическая подготовка по информатике», «Практикум по решению задач по информатике» и учебной дисциплины компонента учреждения образования «Вычислительные методы и компьютерное моделирование».

В результате изучения учебной дисциплины студент должен:

знать:

- основные парадигмы, языки, системы и технологии программирования;
- принципы и технологии структурно-модульного, объектно-ориентированного и визуального программирования;
- методы и этапы разработки и отладки программ;
- возможности применения различных методов и технологий программирования для решения практических задач;

уметь:

- ставить задачу и разрабатывать алгоритм ее решения;
- осуществлять сравнительный анализ языков и технологий программирования;
- разрабатывать объектные модели в различных предметных областях;
- создавать программы на основе объектных моделей;

владеть:

- навыками анализа данных в решаемых задачах и формами их представления;
- способами разработки и отладки программ в современных системах программирования;
- приемами использования современных систем и технологий программирования в профессиональной деятельности учителя информатики.

Освоение учебной дисциплины «Системы и технологии программирования» должно обеспечить формирование у студентов **базовой профессиональной компетенции**: применять методы алгоритмизации в различных средах программирования, систему знаний и навыков из образовательной робототехники для реализации учебного процесса.

На изучение учебной дисциплины «Системы и технологии программирования» отводится всего 108 часов, из которых 50 часов составляют аудиторные. Примерное распределение аудиторных часов по видам занятий: 20 часов – лекции, 30 часов – лабораторные.

Рекомендуемая форма промежуточной аттестации – экзамен.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование раздела, темы	Всего аудиторных часов	из них	
			лекции	лабораторные
Раздел 1	Объектно-ориентированное программирование	26	10	16
Тема 1.1	Системы программирования. Платформа .Net Framework.	6	2	4
Тема 1.2	Основы объектно-ориентированного программирования.	8	4	4
Тема 1.3	Наследование. Полиморфизм.	6	2	4
Тема 1.4	Иерархии классов .Net.	6	2	4
Раздел 2	Технологии визуального программирования	24	10	14
Тема 2.1	Приложения с графическим интерфейсом.	4	2	2
Тема 2.2	Визуальное программирование Windows-приложений.	6	2	4
Тема 2.3	Разработка приложений Windows.Forms.	8	4	6
Тема 2.4	Разработка приложений Windows Presentation Foundation.	4	2	2
	Всего	50	20	30

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

РАЗДЕЛ 1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

Тема 1.1. Системы программирования. Платформа .Net Framework.

Языки, парадигмы и технологии программирования. Структурное, событийно-управляемое, объектно-ориентированное программирование. Системы программирования. Состав и назначение. Системы быстрой разработки программ (RAD). Интегрированные среды разработки программного обеспечения (IDE).

Архитектура платформы .Net Framework. Основные компоненты .Net. Общеязыковая исполнительная среда CLR. Библиотеки классов. Пространства имен. Сборка, решение, проект. Система программирования Microsoft Visual Studio. Работа в среде Visual Studio (Sharp Develop). Основные инструменты. Структурное программирования на языке C#. Реализация базовых алгоритмических конструкций. Обработка исключений. Разработка, отладка и документирование программ.

Тема 1.2. Основы объектно-ориентированного программирования.

Понятие класса и объекта. Состав класса. Модификаторы элементов класса. Поля. Методы. Создание и инициализация объекта. Конструкторы. Параметры методов. Вызов методов, способы передачи параметров. Перегрузка методов. Организация доступа к полям класса. Свойства. Инкапсуляция. Соккрытие полей, создание свойств. Статические классы. Статические поля и методы.

Принципы ООП. Особенности реализация ООП в среде Microsoft Visual Studio. Этапы разработки программ. Визуальное представление классов. Элементы UML. Конструирование классов с помощью визуального дизайнера.

Тема 1.3. Наследование. Полиморфизм.

Разновидности наследования. Классическое наследование. Создание классов-потомков. Конструкторы объектов при наследовании. Модель включения-делегирования. Стратегии наследования. Раннее и позднее связывание. Реализации полиморфизма. Использование виртуальных методов. Переопределение методов. Бесплодные классы.

Тема 1.4. Иерархии классов .Net.

Класс System.Object. Методы класса Object. Абстрактные классы. Интерфейсы. Реализации интерфейсов. Особенности наследования интерфейсов. Стандартные интерфейсы .Net. Делегаты. События.

Тип данных Struct. Перечисления. Класс Enum. Абстрактные структуры данных. Коллекции. Класс ArrayList. Обобщенные типы данных. Использование обобщенных типов и методов. Классы List<T>, Dictionary<K,T>. Достоинства и недостатки ООП.

РАЗДЕЛ 2. ТЕХНОЛОГИИ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ

Тема 2.1. Приложения с графическим интерфейсом.

Интерфейс пользователя (UI). Виды UI. Текстовый интерфейс. Графический интерфейс пользователя (GUI). Оконный интерфейс. Средства программирования интерфейсов. API. Библиотеки для создания приложений с графическим интерфейсом. Основные принципы проектирования GUI. Библиотеки элементов интерфейса Windows-приложений.

Тема 2.2. Визуальное программирование Windows-приложений.

Объектно-событийная модель работы Windows-приложения. Этапы создания Windows-приложения в среде Microsoft Visual Studio. Структура проекта. Разделение визуального представления и функциональности. Реализация функционала.

Элементы управления Windows.Forms. Класс Control. Свойства, методы, события. Класс Form. Свойства, методы, события. Использование элементов TextBox, Button, Label, CheckBox, RadioButton, ListBox, ComboBox, TrackBar, NumericUpDown. Обработка событий. Контейнеры. Позиционирование элементов. PictureBox. Основные свойства и методы. События мыши и клавиатуры.

Тема 2.3. Разработка приложений Windows.Forms.

Разработка простого текстового редактора. Проектирование графического интерфейса. Организация меню. Виды окон. Диалоговые окна. ColorDialog. FontDialog. Окно сообщений MessageBox. Работа с папками и файлами. OpenFileDialog. SaveFileDialog. Работа с датой и временем. Часы и секундомер. DateTimePicker. Класс Timer. Анимация. Работа с базами данных.

Работа с графикой в .Net. Классы и методы рисования. Класс Graphic. Классы Pen, Brush, Color, Font. Векторная графика. Графические примитивы. Рисование фигур. Разработка простого графического редактора. Обработка событий мыши и клавиатуры. Работа с растровыми изображениями. Класс Bitmap. Визуализация данных. Построение диаграмм и графиков.

Тема 2.4. Разработка приложений Windows Presentation Foundation.

Развитие технологий визуального программирования Windows приложений. Визуальные возможности технологий Windows Presentation Foundation (WPF). Унификация интерфейсов. Основы языка разметки XAML. Объектно-событийная модель. Интерактивные возможности WPF. Элементы управления. Привязка данных. Стили. Шаблоны. Ресурсы. Графические возможности WPF. Работа со шрифтами. Работа с векторной и растровой графикой. Анимация. Звук и видео.

Использование возможностей среды MS Visual Studio для разработки информационных систем и образовательных средств. Разработка простых компьютерных моделей и демонстраций. Системы тестирования.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

ОСНОВНАЯ ЛИТЕРАТУРА

1. Городко, С. И. Современные технологии программирования : учеб.-метод. пособие для студентов / С. И. Городко, С. В. Снисаренко; – Минск : БГУИР, 2017. – 67 с.
2. Заборовский, Г.А. Программирование на языке С# / Г.А. Заборовский, В.В. Сидорик, – Минск, БНТУ, 2020. – 83 с.
3. Информатика: учеб. пособие для 11 кл. учреждений общ. сред. образования с рус. яз. обучения / В. М. Котов [и др.]. – Минск : Нар. асвета, 2021. – 114 с.

ДОПОЛНИТЕЛЬНАЯ ИТЕРАТУРА

4. Албахари, Дж, С# 9.0. Справочник. Полное описание языка. – М.: Диалектика, 2021. – 1056 с.
5. Васильев, А.Н. Программирование на С# для начинающих. Особенности языка. -- М: Эксмо, 2019. – 528 с.
6. Гриффитс, И. Програмируем на С# 8.0. Разработка приложений. – СПб.: Питер, 2021. – 944 с.
7. Подбельский, В.В. Язык декларативного программирования XAML. – М.: ДМК, 2018. – 336 с.
8. Прайс, М. С# 9 и .NET 5. Разработка и оптимизация. – СПб.: Питер, 2022. – 832 с.
9. Фленов, М. Библия С#. – СПб.: БХВ-Петербург, 2016. – 544 с.

РЕКОМЕНДУЕМЫЕ ФОРМЫ И МЕТОДЫ ОБУЧЕНИЯ

Для освоения данной учебной дисциплины предусмотрены следующие формы учебных занятий: лекции, в ходе которых излагается теоретический материал; лабораторные занятия – их целью является формирование практических умений и навыков применять теоретические знания; самостоятельная работа, подразумевающая изучение студентами основной и дополнительной литературы.

На лекциях особое внимание уделяется использованию цифровых и мультимедийных технологий для демонстрации особенностей и возможностей изучаемых систем и методов программирования, а также для оперативного получения обратной связи.

Лабораторные занятия направлены на закрепление теоретического материала и формирование умений и навыков практического использования полученных знаний при выполнении практико-ориентированных заданий. Организация лабораторных работ предполагает использование лично-ориентированных методов обучения, что способствует развитию индивидуально-творческих способностей каждого студента и приобретению навыков самостоятельной работы. Повышению эффективности лабораторных занятий способствует использование современных сетевых средств управления учебным компьютерным классом. Для самостоятельной работы рекомендуется использовать интерактивные учебные пособия, тренажеры, тестирующие программы. Важную роль для достижения целей обучения в рамках учебной дисциплины играет исследовательская работа студентов. Эта работа должна органично включаться в учебный процесс со всеми видами занятий.

Рекомендуемые методы обучения: методы проблемного обучения (проблемное изложение, частично-поисковый и исследовательский методы), интерактивные методы и метод проектов, которые способствуют поддержанию оптимального уровня активности, а также эвристическая беседа, исследовательский метод, консультирование, самостоятельная работа студентов.

Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Содержание и формы самостоятельной работы студентов разрабатываются в соответствии с целями и задачами подготовки специалиста.

Для управления самостоятельной работой рекомендуется использовать:

- электронные средства обучения (мультимедийные презентации, электронные пособия);
- материалы интерактивного электронного учебно-методического

комплекса университета;

- материалы ресурсного центра физико-математического факультета;
- тестирующие программы.

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы студентов:

- подготовка к лабораторным работам;
- выполнение примеров и заданий лабораторных работ;
- оформление отчетов по лабораторным работам, презентация результатов;
- выполнение заданий для самостоятельной работы по темам учебной дисциплины;
- выполнение учебно-исследовательской работы.

Текущий контроль осуществляется в ходе выполнения и защиты лабораторных работ, проектов. Самостоятельная работа студента организуется путем выполнения домашних заданий по материалу, пройденному на практических и лабораторных занятиях.

Особое внимание необходимо обращать на организацию индивидуальной работы студента под руководством преподавателя. Эта работа должна проводиться с учетом индивидуальных особенностей каждого студента с помощью системы индивидуальных заданий, которые студент может выполнять на основе образцов, рассмотренных на лекциях.

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ РЕЗУЛЬТАТОВ УЧЕБНОЙ ДЕЯТЕЛЬНОСТИ

Основными средствами диагностики усвоения знаний, умений и овладения необходимыми навыками по учебной дисциплине являются:

- фронтальный опрос на лекционных занятиях, направлен на систематизацию знаний студентов, определение уровня готовности аудитории к восприятию нового материала, а также на формирование у преподавателя представление об усвоении студентами основополагающих понятий и фактов изучаемой учебной дисциплины;

- проверка практических заданий (репродуктивные, продуктивные, творческие задания), выполняемых на лабораторных занятиях, представляет собой диагностику систематичности подготовки студентов к занятиям и уровня усвоения ими практико-ориентированного содержания программного материала учебной дисциплины;

- групповые и индивидуальные консультации студентов, которые предназначены для диагностики уровня овладения знаниями, умениями и навыками, устранения возможных ошибок, пробелов в знаниях студентов;

- самостоятельные работы используются для определения индивидуальных особенностей, темпа продвижения студентов и усвоения ими необходимых знаний;

– компьютерное тестирование позволяет быстро провести диагностику усвоения студентами учебного материала как по отдельным темам и разделам учебной дисциплины, так и по учебной дисциплине в целом.

-- экзамен используется для осуществления итоговой диагностики усвоения учащимися содержания учебной дисциплины за учебный семестр с оценкой в соответствии с критериями оценки результатов учебной деятельности обучающихся в учреждениях высшего образования.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ВОПРОСОВ К ЭКЗАМЕНУ

1. Парадигмы, языки и технологии программирования.
2. Системы программирования. Компоненты. Системы быстрой разработки программ (RAD). Среды визуального проектирования. IDE.
3. Платформа .Net Framework. Архитектура платформы .Net Framework. Основные компоненты. Библиотеки классов .Net.
4. Общеязыковая исполнительная среда CLR платформы .Net Framework.
5. Разработка приложений на платформе .NET. Сборка, решение, проект. Пространство имен.
6. Система программирования Microsoft Visual Studio.
7. Структурное программирование на языке C#. Реализация базовых алгоритмических конструкций.
8. Исключительные ситуации. Обработка исключений.
9. Основы объектно-ориентированного программирования. Понятие класса и объекта. Создание класса. Модификаторы доступа.
10. Состав класса. Поля. Методы. Модификаторы элементов класса. Статические классы. Статические поля и методы.
11. Методы. Вызов методов. Способы передачи параметров.
12. Перегрузка методов. Полиморфизм.
13. Создание и инициализация объекта. Конструкторы.
14. Организация доступа к полям класса в C#. Свойства. Инкапсуляция.
15. Наследование. Разновидности наследования. Классическое наследование.
16. Наследование. Модель включения – делегирования.
17. Стратегии наследования. Виртуальные методы.
18. Реализация полиморфизма. Переопределение методов.
19. Наследование. Использование виртуальных методов.
20. Абстрактные классы. Абстрактные методы. Бесплодные классы.
21. Иерархия классов в C#. Класс Object. Основные методы.
22. Интерфейсы. Реализации интерфейсов. Стандартные интерфейсы .Net.
23. Делегаты. События.
24. Коллекции. Класс ArrayList.
25. Обобщенные типы данных. Использование обобщенных типов и методов. Классы List<T>, Dictionary<K,T>.
26. Визуальное представление классов. Элементы UML.
27. Проектирование программных интерфейсов. Интерфейс пользователя (UI). Основные средства и методы. Виды UI.

28. Текстовый пользовательский интерфейс. Графический интерфейс пользователя (GUI). Оконный интерфейс.
29. Средства программирования интерфейсов. Windows API.
30. Проектирование программных интерфейсов. Windows Forms (WF)
31. Windows-приложения с графическим интерфейсом. Объектно-событийная модель работы Windows-приложения.
32. Элементы управления Windows Forms.
33. Класс Control. Основные свойства и события.
34. Класс Form. Свойства, методы, события.
35. Создание приложения Windows Forms. Структура проекта.
36. Создание приложения Windows Forms. Объектно-событийная модель.
37. Обработка событий мыши. Обработка событий клавиатуры.
38. Элементы управления Windows Forms. TextBox, Button, Label.
39. CheckBox, RadioButton, ListBox. TrackBar, NumericUpDown,
40. Контейнеры. Позиционирование элементов. Использование элемента PictureBox.
41. Класс DateTime. Часы. Класс Timer. Анимация.
42. Создание приложения Windows Forms. Виды окон. Диалоговые окна. Окно сообщений MessageBox. ColorDialog. FontDialog.
43. Диалоговые окна. OpenFileDialog. SaveFileDialog. DateTimePicker.
44. Создание приложения Windows Forms. Организация меню.
45. Работа с графикой GDI+. Классы и методы рисования. Класс Graphics. Графические примитивы.
46. Классы Pen, Brush, Color, Font. Рисование фигур. Обработка растровых изображений.
47. Разработка приложений Windows Presentation Foundation.
48. Интерактивные возможности WPF. Элементы управления. Привязка данных.