

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учебно-методическое объединение по образованию
в области информатики и радиоэлектроники

УТВЕРЖДАЮ

Первый заместитель Министра образования
Республики Беларусь

_____ А.Г. Баханович

_____ /тип.
Регистрационный № ТД-_____ /тип.

**ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ
ПРОГРАММИРОВАНИЯ И СТАНДАРТЫ ПРОЕКТИРОВАНИЯ**

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

1-40 01 01 Программное обеспечение информационных технологий

СОГЛАСОВАНО

Председатель Учебно-методического
объединения по образованию в
области информатики и
радиоэлектроники

_____ В.А. Богуш

СОГЛАСОВАНО

Начальник Главного управления
профессионального образования
Министерства образования
Республики Беларусь

_____ С.Н. Пищов

СОГЛАСОВАНО

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

_____ И.В. Титович

_____ /тип.
Эксперт-нормоконтролер

Минск 2023

СОСТАВИТЕЛЬ:

Л.В.Серебряная, доцент кафедры программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники», кандидат технических наук, доцент;

К.А.Сурков, старший преподаватель кафедры программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

РЕЦЕНЗЕНТЫ:

Кафедра технологий программирования Белорусского государственного университета (протокол № 8 от 19.01.2023);

А.В.Остапенко, профессор кафедры информационных технологий и математики учреждения образования «БИП – университет права и социально-информационных технологий», кандидат физико-математических наук

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ В КАЧЕСТВЕ ТИПОВОЙ:

Кафедрой программного обеспечения информационных технологий учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 11 от 23.01.2023);

Научно-методическим советом учреждения образования «Белорусский государственный университет информатики и радиоэлектроники» (протокол № 7 от 17.03.2023);

Научно-методическим советом по разработке программного обеспечения и информационно-коммуникационным технологиям Учебно-методического объединения по образованию в области информатики и радиоэлектроники (протокол № 7 от 13.03.2023)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ХАРАКТЕРИСТИКА УЧЕБНОЙ ДИСЦИПЛИНЫ

Типовая учебная программа по учебной дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования» разработана для студентов учреждений высшего образования, обучающихся по специальности 1-40 01 01 «Программное обеспечение информационных технологий» в соответствии с требованиями образовательного стандарта высшего образования I степени и типового учебного плана вышеуказанной специальности.

Учебная дисциплина «Объектно-ориентированные технологии программирования и стандарты проектирования» предполагает расширение и углубление знаний будущего инженера-программиста в направлении расширяемого и сборочного программирования, которое существенно повышает производительность труда программиста.

Объектно-ориентированные технологии и стандарты как методы расширяемого и сборочного программирования применяются во всех современных библиотеках и платформах программирования. Функции, выполняемые современными программными средствами (ПС), становятся все более сложными и разнообразными. С учетом этого растут размеры и сложность разрабатываемых ПС, возрастают требования к их качеству. Создание таких ПС без знания и использования основ организации процесса их разработки, а также современных методологий, технологий и инструментальных средств разработки ПС, является невозможным. Это определило место и актуальность данной учебной дисциплины в основе формирования знаний и навыков современного специалиста в области программного обеспечения информационных технологий

Воспитательное значение учебной дисциплины «Объектно-ориентированные технологии программирования и стандарты проектирования» заключается в формировании у обучающихся математической культуры и научного мировоззрения; развитии исследовательских умений, аналитических способностей, креативности, необходимых для решения научных и практических задач; развитии познавательных способностей и активности: творческой инициативы, самостоятельности, ответственности и организованности; формировании способностей к саморазвитию, самосовершенствованию и самореализации.

Изучение данной учебной дисциплины способствует созданию условий для формирования интеллектуально развитой личности обучающегося, которой присущи стремление к профессиональному совершенствованию, активному участию в экономической и социально-культурной жизни страны, гражданская ответственность и патриотизм.

ЦЕЛЬ, ЗАДАЧИ УЧЕБНОЙ ДИСЦИПЛИНЫ

Цель учебной дисциплины: приобретение теоретических знаний и практических навыков в технологиях, языках и инструментальных средствах объектно-ориентированного программирования, современных методологиях и стандартах проектирования программных средств, позволяющих разрабатывать сложные программные средства высокого качества и существенно повысить производительность труда программиста.

Задачи учебной дисциплины:

освоение базовых знаний и навыков в области методологий и технологий объектно-ориентированного программирования;

приобретение навыков в области реализации типовых приемов объектно-ориентированного подхода к созданию программного обеспечения;

изучение стандартов моделирования и проектирования программного обеспечения;

приобретение базовых знаний в области инструментальных средств разработки программного обеспечения и освоение навыков их применения.

Базовыми учебными дисциплинами по курсу «Объектно-ориентированные технологии программирования и стандарты проектирования» являются «Основы алгоритмизации и программирования», «Алгоритмы и структуры данных». В свою очередь учебная дисциплина «Объектно-ориентированные технологии программирования и стандарты проектирования» является базой для такой учебной дисциплины, как «Базы данных», а также учебных дисциплин компонента учреждения высшего образования «Проектирование информационных систем», «Современные платформы программирования», «Программное обеспечение цифрового проектирования».

ТРЕБОВАНИЯ К УРОВНЮ ОСВОЕНИЯ СОДЕРЖАНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

В результате изучения учебной дисциплины «Объектно-ориентированные технологии программирования и стандарты проектирования» формируется следующая базовая профессиональная компетенция: использовать объектно-ориентированный подход в технологии разработки программных систем.

В результате изучения учебной дисциплины студент должен:

знать:

технологии объектно-ориентированного программирования;

понятия объектно-ориентированного программирования: класс, объект, метод, виртуальный метод, конструктор, деструктор, свойство, событие, делегат, интерфейс, метакласс, атрибут;

парадигмы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм;

механизм вызова виртуальных методов и механизм вызова методов через интерфейс;

понятие исключительной ситуации, способы обработки исключительных ситуаций и защиты ресурсов от утечки в случае возникновения исключительных ситуаций;

представление понятий объектно-ориентированного программирования в современных языках программирования;

реализацию понятий объектно-ориентированного программирования через более простые понятия процедурного программирования;

принципы модульного расширяемого программирования и реализацию этих принципов в современных языках программирования;

типовые приемы объектно-ориентированного проектирования и программирования;

современные объектно-ориентированные библиотеки, предназначенные для построения пользовательских интерфейсов;

технологии визуального проектирования программ и компонентного программирования;

представление концепций объектно-ориентированного программирования в различных языках программирования;

принципы и особенности автоматизированной разработки программных средств;

уметь:

разрабатывать программы с объектно-ориентированным дизайном;

проектировать программы с использованием понятий объектно-ориентированного программирования;

применять типовые приемы объектно-ориентированного программирования;

разрабатывать надежные программы с защитой от исключительных ситуаций;

разрабатывать модульные расширяемые программы;

разрабатывать программы с использованием современных объектно-ориентированных библиотек;

проектировать программные компоненты с помощью инструментальных средств визуального проектирования.

оценивать эффективность структуры программного средства;

владеть:

навыками применения различных шаблонов проектирования при разработке объектно-ориентированных программных средств;

навыками формализации предметной области с помощью средств объектно-ориентированного анализа;

навыками работы с современными средствами разработки программных средств.

Типовая учебная программа рассчитана на 240 учебных часов, из них – 116 аудиторных. Примерное распределение аудиторных часов по видам занятий: лекции – 60 часов, лабораторные занятия – 56 часов.

ПРИМЕРНЫЙ ТЕМАТИЧЕСКИЙ ПЛАН

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 1. Объектно-ориентированные технологии программирования	64	32	32
Тема 1. Объект как динамический модуль. Исключение. Класс и объект	2	2	-
Тема 2. Свойство. Расширение класса	6	2	4
Тема 3. Виртуальный метод и виртуальное свойство	6	2	4
Тема 4. Делегат. Событие. Метакласс	6	2	4
Тема 5. Интерфейс. Шаблон	6	2	4
Тема 6. Атрибут. Анонимная функция	6	2	4
Тема 7. Приемы объектно-ориентированного программирования	8	4	4
Тема 8. Визуальное компонентное и сборочное программирование	2	2	-
Тема 9. Объектно-ориентированное программирование в языке Java	6	2	4
Тема 10. Имитация модульного программирования в языке C++	6	2	4
Тема 11. Классы и объекты в языке C++. Исключения в языке C++	2	2	-
Тема 12. Множественное наследование в языке C++	2	2	-
Тема 13. Виртуальные методы в языке C++. Константные методы	2	2	-
Тема 14. Операторы приведения типа в языке C++. Ссылки в языке C++. Перегрузка операторов в языке C++	2	2	-
Тема 15. Шаблоны функций и классов в языке C++	2	2	-
Раздел 2. CASE-технологии объектно-ориентированного анализа и проектирования программных средств	10	10	-
Тема 16. CASE-технологии	2	2	-
Тема 17. Критерии развития CASE-средств	2	2	-
Тема 18. Обзор современных объектно-ориентированных программных средств	2	2	-
Тема 19. Типы и особенности современных программных проектов	2	2	-
Тема 20. Особенности построения объектно-ориентированного программного средства	2	2	-

Наименование раздела, темы	Всего аудиторных часов	Лекции	Лабораторные занятия
Раздел 3. Стандарты проектирования на унифицированном языке моделирования UML	30	10	20
Тема 21. Общие сведения об унифицированном языке моделирования UML	2	2	-
Тема 22. Создание проекта на UML	22	6	16
Тема 23. Автоматизация кодогенерации проекта и создание приложения на его основе	6	2	4
Раздел 4. Дополнительные возможности объектно-ориентированных CASE-средств	6	2	4
Тема 24. Моделирование Web-приложений в CASE-среде	6	2	4
Раздел 5. Унифицированный процесс разработки объектно-ориентированных ПС (RUP). Объектно-ориентированные технологии создания программных средств	6	6	-
Тема 25. Унифицированный процесс разработки объектно-ориентированных ПС	2	2	-
Тема 26. Аспекты RUP	2	2	-
Тема 27. Современные технологии создания программных средств с применением средств автоматизации	2	2	-
Итого:	116	60	56

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

Тема 1. ОБЪЕКТ КАК ДИНАМИЧЕСКИЙ МОДУЛЬ. ИСКЛЮЧЕНИЕ. КЛАСС И ОБЪЕКТ.

Понятие модуля. Принципы модульного программирования. Разграничение доступа к элементам модуля. Термин «инкапсуляция». Понятие объекта как динамического модуля. Понятие класса как описания динамического модуля.

Понятия ошибки и исключения. Создание исключений. Классы исключений. Обработка исключений. Защита ресурсов от утечки в случае исключений. Приемы надежного программирования.

Понятие класса. Создание объекта. Понятие метода. Представление метода в виде обычной процедуры. Понятие конструктора и деструктора.

Тема 2. СВОЙСТВО. РАСШИРЕНИЕ КЛАССА.

Понятие свойства как виртуального поля. Назначение свойства. Методы получения и установки значений свойства. Режимы доступа к свойству. Свойство-индексатор или свойство-массив (в зависимости от языка программирования).

Расширение класса путем создания производного класса. Термин «наследование». Существование «прародителя» всех классов. Перекрытие элементов класса в производных классах. Совместимость объектов различных классов. Контроль и преобразование типов.

Тема 3. ВИРТУАЛЬНЫЙ МЕТОД И ВИРТУАЛЬНОЕ СВОЙСТВО

Понятие виртуального метода. Перекрытие виртуального метода в производном классе. Абстрактный виртуальный метод. Механизм вызова виртуального метода. Динамический виртуальный метод (в некоторых языках программирования). Метод обработки сообщений (в некоторых языках программирования). Термин «полиморфизм». Понятие виртуального свойства. Запрет на расширение класса.

Тема 4. ДЕЛЕГАТ. СОБЫТИЕ. МЕТАКЛАСС.

Понятие делегата (ссылки на метод объекта – в зависимости от языка программирования). Пример применения делегата. Понятие события как списка делегатов. Пример применения события. Методы регистрации события.

Понятие метакласса (в некоторых языках программирования). Методы, применяемые к классам. Виртуальные конструкторы (в некоторых языках).

Тема 5. ИНТЕРФЕЙС. ШАБЛОН.

Понятие интерфейса. Описание интерфейса. Поддержка интерфейса классом. Механизм подсчета ссылок в интерфейсах. Расширение интерфейса. Глобально-уникальный идентификатор интерфейса. Совместимость

интерфейсов и классов. Получение интерфейса через другой интерфейс. Представление интерфейса в памяти. Механизм вызова метода объекта через интерфейс. Применение интерфейса для доступа к объекту динамически подключаемой библиотеки.

Понятие шаблона как параметризованного класса. Создание переменной на базе шаблонного типа. Представление данных, заданных шаблоном, в памяти. Параметры шаблона. Установка ограничений на параметры шаблона. Шаблон делегата.

Тема 6. АТТРИБУТ. АНОНИМНАЯ ФУНКЦИЯ.

Информация о классах во время выполнения программы. Понятие «рефлексии» объектов. Понятие атрибута (аннотации – в другом языке программирования) как механизма рефлексии. Создание пользовательского атрибута. Проверка атрибута во время выполнения программы. Пример применения атрибутов.

Варианты анонимной функции. Шаблон делегата-функции. Представление анонимной функции в виде данных. Ограничения на выражения внутри анонимной функции.

Тема 7. ПРИЕМЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

Итератор. Одиночка. Заместитель. Компоновщик. Мост. Наблюдатель. Посетитель. Фабричный метод. Фабрика классов. Пул объектов.

Тема 8. ВИЗУАЛЬНОЕ КОМПОНЕНТНОЕ И СБОРОЧНОЕ ПРОГРАММИРОВАНИЕ

Понятие компонента. Понятие визуального программирования. Инструментальные средства визуального компонентного программирования. Современные библиотеки компонентов. Понятие расширяемой программы и расширяемого программирования. Принципы расширяемого программирования. Применение методологии объектно-ориентированного программирования для построения расширяемых программ.

Тема 9. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ В ЯЗЫКЕ JAVA

Модуль, класс, объект, метод, атрибуты доступа к элементам объекта, интерфейс, шаблон, аннотация. Отличия средств объектно-ориентированного программирования в языках C# и Java.

Тема 10. ИМИТАЦИЯ МОДУЛЬНОГО ПРОГРАММИРОВАНИЯ В ЯЗЫКЕ C++ Имитация модульного программирования в языке C++. Пространство имен.

Тема 11. КЛАССЫ И ОБЪЕКТЫ В ЯЗЫКЕ C++. ИСКЛЮЧЕНИЯ В ЯЗЫКЕ C++
Классы в языке C++. Наследование. Конструкторы и деструкторы. Стандартные конструкторы. Создание объектов по значению (на стеке) и по

ссылке (в динамической памяти). Операторы `new` и `delete`. Размещающий оператор `new`. Порядок конструирования и разрушения объектов. Вложенные определения классов. «Друзья» класса. Статические члены класса.

Обработка исключений ситуаций в языке C++. Защита от утечки ресурсов. Имитация оператора `try-finally`. Понятие автоматического указателя (`auto_ptr`). Использование автоматических указателей для защиты от утечки ресурсов.

Тема 12. МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ В ЯЗЫКЕ C++

Множественное наследование. Проблема повторяющихся базовых классов. Замена множественного наследования наследованием от интерфейсов в других языках объектно-ориентированного программирования.

Тема 13. ВИРТУАЛЬНЫЕ МЕТОДЫ В ЯЗЫКЕ C++. КОНСТАНТНЫЕ МЕТОДЫ

Виртуальные методы в языке C++. Недостатки синтаксиса определения и перекрытия виртуальных методов в языке C++. Понятие константного метода. Проблемы, порождаемые наличием константных методов.

Тема 14. ОПЕРАТОРЫ ПРИВЕДЕНИЯ ТИПА В ЯЗЫКЕ C++. ССЫЛКИ В ЯЗЫКЕ C++. ПЕРЕГРУЗКА ОПЕРАТОРОВ В ЯЗЫКЕ C++

Операторы приведения типа в языке C++: `const_cast`, `reinterpret_cast`, `static_cast`, `dynamic_cast`.

Ссылки в языке C++. Рекомендации по работе со ссылками. Типичные ошибки при работе со ссылками.

Перегрузка бинарных операторов. Перегрузка унарных операторов. Перегрузка операторов преобразования типа.

Тема 15. ШАБЛОНЫ ФУНКЦИЙ И КЛАССОВ В ЯЗЫКЕ C++

Шаблоны функций в языке C++. Шаблоны классов в языке C++. Специализации шаблонов классов. Проблемы шаблонов в языке C++.

Раздел 2. CASE-ТЕХНОЛОГИИ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО АНАЛИЗА И ПРОЕКТИРОВАНИЯ ПРОГРАММНЫХ СРЕДСТВ

Тема 16. CASE-ТЕХНОЛОГИИ

История появления и особенности развития. Системная модель CASE-средств: обозначение, наименование, цели, общесистемные характеристики и структура.

Тема 17. КРИТЕРИИ РАЗВИТИЯ CASE-СРЕДСТВ

Функциональные, технологические, экономические и эргономические критерии CASE-средств.

Тема 18. ОБЗОР СОВРЕМЕННЫХ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПРОГРАММНЫХ СРЕДСТВ

Назначение, особенности работы и функциональные возможности программ Rational Rose, Rational XDE, Enterprise Architect.

Тема 19. ТИПЫ И ОСОБЕННОСТИ СОВРЕМЕННЫХ ПРОГРАММНЫХ ПРОЕКТОВ

Три составляющие программного проекта: система обозначений, процесс и инструмент. Их роль и значение для проекта. Целесообразность использования различных методологий для различных типов программных проектов.

Тема 20. ОСОБЕННОСТИ ПОСТРОЕНИЯ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММНОГО СРЕДСТВА

Объектно-ориентированная технология. Основные и дополнительные элементы объектной модели. Сравнительный анализ с моделями других типов.

РАЗДЕЛ 3. СТАНДАРТЫ ПРОЕКТИРОВАНИЯ НА УНИФИЦИРОВАННОМ ЯЗЫКЕ МОДЕЛИРОВАНИЯ UML

Тема 21. ОБЩИЕ СВЕДЕНИЯ ОБ УНИФИЦИРОВАННОМ ЯЗЫКЕ МОДЕЛИРОВАНИЯ UML

Особенности, цели создания и нотация языка UML.

Тема 22. СОЗДАНИЕ ПРОЕКТА НА UML

Назначение, синтаксис, инструменты диаграмм вариантов использования и анализа устройств на языке UML.

Назначение, синтаксис, инструменты диаграмм состояний и деятельности на языке UML.

Назначение, синтаксис, инструменты диаграмм взаимодействия и сотрудничества на языке UML. Шаблоны проектирования на основе стандарта UML.

Назначение, синтаксис, инструменты диаграмм компонентов и классов на языке UML.

Тема 23. АВТОМАТИЗАЦИЯ КОДОГЕНЕРАЦИИ ПРОЕКТА И СОЗДАНИЕ ПРИЛОЖЕНИЯ НА ЕГО ОСНОВЕ

Автоматизация создания кода класса. Обновление кода по модели и модели по коду. Структура приложения и автоматизация создания его шаблона.

Раздел 4. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ CASE-СРЕДСТВ

Тема 24. МОДЕЛИРОВАНИЕ WEB-ПРИЛОЖЕНИЙ В CASE-СРЕДЕ

Web-стереотипы. Элементы Web-приложения и этапы его создания.

Раздел 5. УНИФИЦИРОВАННЫЙ ПРОЦЕСС РАЗРАБОТКИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПС (RUP). ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНЫХ СРЕДСТВ

Тема 25. УНИФИЦИРОВАННЫЙ ПРОЦЕСС РАЗРАБОТКИ ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ ПС

Особенности унифицированного процесса разработки RUP. Основные этапы RUP. Артефакты и прецеденты.

Тема 26. АСПЕКТЫ RUP

Статический и динамический аспекты RUP, потоки и процессы RUP. Использование RUP в сочетании с языком UML.

Тема 27. СОВРЕМЕННЫЕ ТЕХНОЛОГИИ СОЗДАНИЯ ПРОГРАММНЫХ СРЕДСТВ С ПРИМЕНЕНИЕМ СРЕДСТВ АВТОМАТИЗАЦИИ

Принципы и стадии разработки программных средств в различных технологиях. Содержание и результаты каждой стадии разработки.

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ**ЛИТЕРАТУРА****ОСНОВНАЯ**

1. Сурков, К. А. Объектно-ориентированные технологии программирования [Электронный ресурс] / К. А. Сурков, Д. А. Сурков, Ю. М. Четырько. – Режим доступа: <https://nezaboodka.by/assets/docs/OOP.2021-09-08.pdf>. – Дата доступа: 14.03.2023
2. Полное руководство по языку программирования C# 11 и платформе .NET 7 [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/tutorial/>. – Дата доступа: 14.03.2023
3. Троэлсен, Э. Язык программирования C# 7 и платформы .NET и .NET Core / Э. Троэлсен, Ф. Джепикс – 8-е изд. – пер. с англ. – Санкт-Петербург : Диалектика, 2019. – 370 с.
4. Скит, Дж. C# для профессионалов: тонкости программирования / Дж. Скит – 3-е изд. – пер. с англ. – Москва : Вильямс, 2014. – 608 с.
5. Эккель, Б. Философия Java / Б. Эккель – 4-е полное изд. – пер. с англ. – Санкт-Петербург : Питер, 2018. – 1168 с.
6. Шилдт, Г. Java 8. Полное руководство / Г. Шилдт. – 9-е изд. – пер. с англ. – Москва : Вильямс, 2015. – 816 с.
7. Шилдт, Г. C++ для начинающих / Г. Шилдт – пер. с англ. – Москва : ЭКОМ Паблишерз, 2013. – 640 с.
8. Прата, Ст. Язык программирования C++. Лекции и упражнения / Ст. Прата – 6-е изд. – пер. с англ. – Москва : Вильямс, 2018. – 928 с.
9. Лафоре, Р. Объектно-ориентированное программирование в C++ / Р. Лафоре. – 4-е изд. – пер. с англ. – Санкт-Петербург : Питер, 2019. – 928 с.
10. Крачтен, Ф. Введение в Rational Unified Process / Ф. Крачтен. – Москва : Вильямс, 2002. – 240 с.
11. Поллис, Г. Разработка программных проектов на основе Rational Unified Process (RUP) / Г. Поллис, Л. Огастин. – Москва : Бином, 2005. – 113 с.
12. Фаулер, М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования / М. Фаулер. – Москва : Символ-Плюс, 2011. – 192 с.
13. Трофимов, С. А. CASE-технологии: практическая работа в Rational Rose / С. А. Трофимов. – Москва : Бином-Пресс, 2002. – 288 с.
14. Кватрани, Т. Визуальное моделирование с помощью Rational Rose 2002 и UML / Т. Кватрани. – Москва : Вильямс, 2003. – 192 с.
15. Rational XDE / У. Боггс, М. Боггс. – Санкт-Петербург : Лори, 2007. – 580 с.
16. Трофимов, С. А. Rational XDE для Visual Studio .NET / С. А. Трофимов. – Москва : Бином-Пресс, 2004. – 297 с.
17. Васильев, А. Н. Программирование на C++ в примерах и задачах / А. Н. Васильев. – Москва : Эксмо, 2021. – 365 с.
18. Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. – Москва : Юрайт, 2021. – 155 с.

19. Паттерны объектно-ориентированного проектирования : юбилейное издание легендарной книги банды четырех / Э. Гамма [и др.]. – Санкт-Петербург : Питер, 2021. – 446 с.

20. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. – Москва : Юрайт, 2021. – 206 с.

21. Чернышев, С. А. Принципы, паттерны и методологии разработки программного обеспечения : учебное пособие для вузов / С. А. Чернышев. – Москва : Юрайт, 2021. – 176 с.

ДОПОЛНИТЕЛЬНАЯ

22. ECMA-334 C# Language Specification [Электронный ресурс]. – Режим доступа: <https://www.ecma-international.org/publications-and-standards/standards/ecma-334/>. – Дата доступа: 14.03.2023

23. International Standard ISO/IEC 14882:2020, Programming languages – C++.

24. Wirth N. The Programming Language Oberon, rev 1.12.2007 [Электронный ресурс]. – Режим доступа: <https://people.inf.ethz.ch/wirth/Oberon/Oberon07.Report.pdf>. – Дата доступа: 14.03.2023

25. Европейский центр программирования [Электронный ресурс]. – Режим доступа: <http://www.europrog.ru/>. – Дата доступа: 14.03.2023

26. Васильев, А. Программирование на C# для начинающих. Основные сведения / А. Васильев. – Москва : Эксмо, 2018. – 592 с.

27. Вальвачев, А. Н. Программирование на языке Delphi : учебное пособие [Электронный ресурс] / А. Н. Вальвачев, К. А. Сурков, Д. А. Сурков, Ю. М. Четырько. – Режим доступа: <http://www.delphikingdom.com/lyceum/seminar.asp?ID=6>. – Дата доступа: 14.03.2023

28. Федотова, Д. З. CASE-технологии : практикум / Д. З. Федотова, Ю. Д. Семенов, К. Н. Чижик. – Москва : Горячая Линия-Телеком, 2005. – 160 с.

29. Калянов, Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов / Г. Н. Калянов. – Москва : Горячая линия-Телеком, 2002. – 320 с.

30. Зиглер, К. Методы проектирования программных средств / К. Зиглер. – Москва : Мир, 1985. – 328 с.

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

При изучении учебной дисциплины рекомендуется использовать следующие формы самостоятельной работы:

решение дополнительных задач повышенной сложности, предлагаемых преподавателем;

участие в конкурсе студенческих работ;

представление результатов самостоятельной работы в рамках студенческих конференций.

ПЕРЕЧЕНЬ РЕКОМЕНДУЕМЫХ СРЕДСТВ ДИАГНОСТИКИ КОМПЕТЕНЦИЙ СТУДЕНТА

Типовым учебным планом по специальности 1-40 01 01 Программное обеспечение информационных технологий в качестве формы промежуточной аттестации по учебной дисциплине «Объектно-ориентированные технологии программирования и стандарты проектирования» рекомендуется экзамен. Оценка учебных достижений студента по итогам зачета производится по десятибалльной шкале.

Для текущего контроля по учебной дисциплине и диагностики компетенций студентов могут использоваться следующие формы:

- отчеты по лабораторным работам;
- контрольные работы;
- устные опросы;
- электронные тесты по разделам учебной дисциплины;
- доклады;
- коллоквиум;
- защита индивидуальных практических работ.

РЕКОМЕНДУЕМЫЕ МЕТОДЫ (ТЕХНОЛОГИИ) ОБУЧЕНИЯ

Основные рекомендуемые методы (технологии) обучения, отвечающие целям и задачам учебной дисциплины:

проблемное обучение (проблемное изложение, вариативное изложение, частично-поисковый метод), реализуемое на лекционных занятиях;

учебно-исследовательская деятельность, творческий подход, реализуемые на лабораторных занятиях;

проектные технологии, используемые при проектировании конкретного объекта, реализуемые при выполнении самостоятельной работы.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ ЛАБОРАТОРНЫХ ЗАНЯТИЙ

1. Изучение иерархий классов и виртуальных методов.
2. Изучение сериализации объектов.
3. Изучение расширяемого программирования.
4. Изучение расширяемого программирования на языке C++.
5. Построение программных компонентов.
6. Создание проекта в среде объектно-ориентированного CASE-средства.
7. Автоматическое построение кода приложения.
8. Реализация функций приложения.
9. Создание Web-приложения в среде объектно-ориентированного CASE-средства.

ПРИМЕРНЫЙ ПЕРЕЧЕНЬ КОМПЬЮТЕРНЫХ ПРОГРАММ
(*необходимого оборудования, наглядных пособий и т. п.*)

1. Операционная система Linux, или MacOS, или Microsoft Windows любой современной версии.
2. Языки и системы программирования: Visual Studio Code (C#, C++), JetBrains Rider (C#), Apache NetBeans (Java), JetBrains IntelliJ IDEA (Java) или Eclipse IDE (Java).
3. CASE-средства: Rational Rose, Rational XDE, Enterprise Architect или другая среда, поддерживающая унифицированный язык моделирования UML и автоматическую кодогенерацию шаблона проекта.