

Белорусский государственный университет



Проректор по учебной работе и
образовательным инновациям

О. Г. Прохоренко

«01» декабря 2022 г.

Регистрационный № УД – 11771/уч.

ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

**Учебная программа учреждения высшего образования
по учебной дисциплине для специальностей:**

1-31 03 05 Актуарная математика

1-31 03 06 Экономическая кибернетика (по направлениям)

направление специальности:

1-31 03 06-01 Экономическая кибернетика (математические
методы и компьютерное моделирование в экономике)

1-98 01 01 Компьютерная безопасность (по направлениям)

направление специальности:

1-98 01 01-01 Компьютерная безопасность (математические
методы и программные системы)

2022 г.

Учебная программа составлена на основе образовательных стандартов высшего образования ОСВО 1-31 03 05-2021, ОСВО 1-31 03 06-2021, ОСВО 1-98 01 01-2021, типовых учебных планов №G 31-1-027/пр-тип от 30.06.2021, №G 31-1-028/пр-тип от 30.06.2021, №P 98-1-003/пр-тип от 02.07.2021, учебных планов №G 31-1-032/уч. от 30.06.2021, №G 31-1-033/уч. от 30.06.2021, №P 98-1-005/уч.от 23.07.2021, №P 98-1-024/уч.ин от 09.08.2021, №G 31-1-215/уч. от 22.03.2022, №P 98-1-206/уч. от 22.03.2022.

СОСТАВИТЕЛИ:

С.Е. Гутников – старший преподаватель кафедры информационных систем управления Белорусского государственного университета

РЕЦЕНЗЕНТЫ:

В.В. Лепин - учёный секретарь института математики НАН Беларуси, кандидат физико-математических наук, доцент

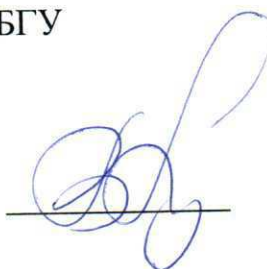
Д.В. Шункевич – заведующий кафедрой интеллектуальных информационных технологий БГУИР, кандидат технических наук, доцент

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой информационных систем управления БГУ
(протокол № 6 от 24.11.2022 г.).

Научно-методическим Советом БГУ
(протокол № 2 от 29.11.2022 г.)

Заведующий кафедрой



В.В. Краснопрошин

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная дисциплина «Технологии программирования» (далее «ТП»), ориентирована на обучение студентов знаниям, умениям и навыкам в области программирования. Изучаемые темы базируются на использовании современных информационных технологий, новейшего программного и технического обеспечения компьютеров.

Особое внимание в учебной дисциплине уделяется методам проектирования программных систем, рассматриваются основные этапы процесса создания программного продукта, в том числе анализ предъявляемых к нему требований, составление технического задания, проектирование, кодирование, включая интеграцию и тестирование программы в целом, а также её сопровождение после установки у заказчика. Отдельные темы посвящены определению типа процесса разработки и управлению проектом, что особенно важно при работе в команде.

При изложении учебной дисциплины «Технологии программирования» важно продемонстрировать комплексный подход к решению основных вопросов, возникающих в процессе разработки программных проектов.

Основой для обучения являются дисциплины модуля «Программирование», изучаемые в 1-м, 2-м и 3-м семестрах: «Основы и методологии программирования», «Разработка кросс-платформенных приложений», «Машинно-ориентированное программирование», «Промышленное программирование».

Цели и задачи учебной дисциплины

Цель преподавания учебной дисциплины «ТП» – освоение основных понятий и методов современной теории и практики создания программных систем. При изложении материала учебной дисциплины важно показать возможности использования методов анализа и проектирования программных систем при решении прикладных задач, возникающих в различных областях науки, техники, экономики и др. Целесообразно выделить моменты анализа математических моделей процессов и задач с целью их последующего проектирования, реализации и тестирования.

Задачи учебной дисциплины

В рамках поставленной цели основные задачи, решаемые при изучении учебной дисциплины «Технологии программирования»:

1. исследование предметной области под названием: «Технологии программирования»;
2. формирование у студентов понятий метод, методология, технология разработки программных систем;
3. изучение методов и технологий разработки программного обеспечения с целью использования при решении практических задач;

4. использование современных инструментов для моделирования, разработки и тестирования программных продуктов.

Место учебной дисциплины в системе подготовки специалиста с высшим образованием.

Учебная дисциплина относится к **модулю** «Программирование» государственного компонента.

Программа составлена с учётом межпредметных **связей** с учебными дисциплинами. Основой для обучения являются дисциплины модуля «Программирование», изучаемые в 1-м, 2-м и 3-м семестрах: «Основы и методологии программирования», «Разработка кросс-платформенных приложений», «Машинно-ориентированное программирование», «Промышленное программирование».

Сформированные при изучении дисциплины «Технологии программирования» компетенции являются основой для дальнейшего изучения дисциплины модуля «Информационные системы»: «Модели данных и СУБД».

Знания, полученные в учебной дисциплине, используются при изучении всех дисциплин специализации, при выполнении курсовых и дипломных работ, а также используются как инструментарий для моделирования и компьютерного решения задач ряда математических дисциплин, изучаемых на старших курсах.

Требования к компетенциям

Освоение учебной дисциплины «Технологии программирования» должно обеспечить формирование следующей **базовой профессиональной компетенции**:

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

В результате освоения учебной дисциплины студент должен:

знать:

- классические и современные модели процесса разработки программных систем;
- принципы, способы и методы разработки и поддержки программных продуктов;
- методы организации и управления процессом разработки ПО;
- стандарты процесса разработки ПО;
- основные способы и методы оценки качества продукта;
- принципы работы с различными инструментами для поддержки процесса разработки программных систем;

уметь:

- моделировать программные системы;
- проводить анализ созданных моделей;

- использовать модели при разработке и поддержке программных систем;

- пользоваться современными CASE-средствами проектирования и тестирования ПО;

владеть:

- основными методами анализа и проектирования ПО;

- методами оценки качества и надежности ПО;

- навыками самообразования и способами использования приемов и методов тестирования в процессе разработки программных продуктов.

Структура учебной дисциплины

Дисциплина изучается в 4-м семестре. Всего на изучение учебной дисциплины «Технологии программирования» отведено

- для очной формы получения высшего образования отведено 108 часов, в том числе 68 аудиторных часа, из них: лекции – 34 часа, лабораторные занятия – 30 часов, управляемая самостоятельная работа – 4 часа.

Трудоёмкость учебной дисциплины составляет 3 зачётных единицы.

Форма текущей аттестации – **зачёт**.

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1. Введение

Тема 1.1. Основные понятия

Сложность. Неудачи IT-проектов. Основные определения. Программное обеспечение (ПО). Технология программирования. Программная инженерия. Жизненный цикл ПО - история появления понятия. Программный продукт (ПП).

Раздел 2. Требования к ПО

Тема 2.1. Определения требований

Классификация требований. Потребности. Бизнес-требования. Пользовательские требования. Функциональные требования. Бизнес-правила. Внешние интерфейсы. Атрибуты качества. Ограничения. Системные требования. Features.

Тема 2.2. Инженерия требований к ПО

Выявление требований. Анализ требований. Спецификация требований. Валидация требований. Верификация требований. Управление требованиями. Участники процессов (Process Actors).

Раздел 3. Проектирование архитектуры ПО

Тема 3.1. Архитектура ПО

Концептуальная архитектура. Архитектурный стиль. Пилотная архитектура. Модуль. Компонент. Фреймворк. Слабая связность. Сквозная функциональность.

Тема 3.2. Принципы качественно спроектированной программной архитектуры

Разделение функций. Принцип основной ответственности. Принцип минимального знания. Отсутствие повторений. Проектирование в процессе разработки ПО. Проектирование и качество ПО. Типы программных приложений.

Тема 3.3. Документирование архитектуры ПО

Цели документирования. Этапы создания документации. Правила документирования программной архитектуры. Нотации для документирования программной архитектуры.

Тема 3.4. Архитектурные представления и стили

Архитектурное представление. Архитектурная модель 4+1. Подход «Views and Beyond». Модульные представления. Наиболее распространенные модульные представления. Компонентные представления. Представления развертывания. Документация архитектуры. Микроархитектура. Внедрение зависимости. Схемы данных. Компонентные архитектурные стили. Сочетание архитектурных стилей. Интеграция приложений.

Раздел 4. Принципы дизайна и парадигмы программирования

Тема 4.1. Парадигмы программирования

Ограничения, связанные с парадигмой программирования. Структурное программирование. Объектно-ориентированное программирование. Функциональное программирование.

Тема 4.2. Принципы дизайна SOLID

Принцип единственной ответственности — Single Responsibility Principle (SRP). Принцип открытости / закрытости — Open-Closed Principle (OCP). Принцип подстановки Барбары Лисков — Liskov Substitution Principle (LSP). Принцип разделения интерфейсов — Interface Segregation Principle (ISP). Принцип инверсии зависимости — Dependency Inversion Principle (DIP).

Раздел 5. Модели жизненного цикла программного продукта

Тема 5.1. Различные модели жизненного цикла программного продукта

Каскадная модель, основные принципы каскадной модели. Преимущества каскадной модели. Недостатки каскадной модели. Усовершенствованная каскадная модель. Спиральная модель. Преимущества спиральной модели. MBASE принципы. Эволюция спиральной модели. Недостатки спиральной модели. V-образная модель. Инкрементная (пошаговая) модель. Модель быстрого прототипирования.

Тема 5.2. Модели промышленных технологий создания программного продукта

Модель Microsoft Solution Framework (MSF). Agile-методологии разработки. Модель MSF for Agile Software Development. Модель Rational Unified Process (RUP). Модель Extreme Programming (XP).

Раздел 6. Unified Modeling Language (UML)

Тема 6.1. Основные компоненты UML

Структура UML. Пакеты в языке UML. Основные пакеты метамодели UML. Описание метамодели языка UML. Диаграммы UML. Особенности изображения диаграмм UML.

Тема 6.2. Диаграмма вариантов использования (Use Case)

Вариант использования. Актеры. Интерфейсы. Примечания. Отношения на диаграмме вариантов использования. Пример построения UseCase-диаграммы.

Тема 6.3. Диаграмма классов

Элементы диаграммы классов. Класс, имя класса, атрибуты класса, операции. Отношения между классами. Интерфейсы. Объекты. Шаблоны или параметризованные классы.

Тема 6.4. Диаграмма состояний

Автоматы. Состояние, имя состояния, список внутренних действий, начальное состояние, конечное состояние, переход, событие. Сторожевое условие. Выражение действия. Составное состояние и подсостояние,

последовательные подсостояния, параллельные подсостояния, историческое состояние. Сложные переходы, переходы между параллельными состояниями, переходы между составными состояниями. Синхронизирующие состояния. Построение диаграмм состояний.

Тема 6.5. Диаграмма деятельности (Activity)

Состояние действия. Переходы. Дорожки. Объекты. Построение диаграмм деятельности.

Тема 6.6. Диаграмма последовательности

Объекты. Линия жизни объекта. Фокус управления. Сообщения. Ветвление потока управления. Временные ограничения. Комментарии или примечания. Построение диаграмм последовательности. Пример диаграммы последовательности.

Тема 6.7. Диаграмма кооперации

Кооперация. Диаграмма кооперации уровня спецификации. Объекты, мультиобъект, активный объект, составной объект. Связи, стереотипы связей. Сообщения, формат записи сообщений. Построение диаграмм кооперации. Пример построения диаграммы кооперации.

Тема 6.8. Диаграммы компонентов и развертывания

Компоненты, имя компонента, виды компонентов. Интерфейсы. Зависимости. Построение диаграмм компонентов. Диаграмма развертывания. Узел. Соединения. Построение диаграмм развертывания.

Раздел 7. Тестирование и отладка ПО

Тема 7.1. Введение в тестирование ПО

Виды тестирования ПО. Экономическая сторона тестирования. Психология тестирования.

Тема 7.2. Модульное тестирование

Модульное тестирование и его задачи. Обзоры программного кода. Тестирования структуры модулей. Тестирование взаимодействия модулей. Стратегии пошагового тестирования. Объектно-ориентированное тестирование. Автоматизация модульного тестирования. Семейство xUnit. Универсальные средства автоматизации тестирования.

Тема 7.3. Функциональное тестирование

Планирование функционального тестирования. Тестовый план. Разработка тестовых случаев. Выбор тестовых данных. Ошибка, свойства ошибки. Составления отчетов об ошибках. Системы документирования/отслеживания ошибок. Жизненный цикл ошибки. Реализация тестов: приемочный тест, позитивный тест, негативный тест. Особенности функционального тестирования.

Тема 7.4. Автоматизация функционального тестирования

Плюсы/минусы автоматизации тестирования. Требования к автоматизированным тестам. Методы автоматизации функционального

тестирования. Метод Play&Record. Метод функциональной декомпозиции. Метод Data-driven. Метод Keyword-driven. Семейство Selenium и его возможности. Проблемы внедрения автоматизации тестирования.

Тема 7.5. Отладка ПО

Методика отладки. Методы отладки. Средства отладки.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Дневная форма получения образования с применением электронных средств обучения (ДО)

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
1	2	3	4	5	6	7	8	9
	Технологии программирования	34			30		4	
1.	Введение	1						
1.1.	Основные понятия	1						
2.	Требования к ПО	2			4			
2.1.	Определения требований	1			2			Отчет по лабораторной работе
2.2.	Инженерия требований к ПО	1			2			Отчет по лабораторной работе
3.	Проектирование архитектуры ПО	4			4			
3.1.	Архитектура ПО	1			2			Отчет по лабораторной работе
3.2.	Принципы качественно спроектированной программной архитектуры	1						
3.3.	Документирование архитектуры ПО	1			2			Отчет по лабораторной работе
3.4.	Архитектурные представления и стили	1						
4.	Принципы дизайна и парадигмы программирования	2						
4.1.	Парадигмы программирования	1						
4.2.	Принципы дизайна SOLID	1						
5.	Модели жизненного цикла программного продукта	4						
5.1.	Различные модели жизненного цикла программного продукта	2						
5.2.	Модели промышленных технологий создания программного продукта	2						Тест по разделам 4-5.
6.	Unified Modeling Language (UML)	16			18		2	
6.1.	Основные компоненты UML	2					2	Собеседование. Контрольная работа №1 (по разделам 1-5)

6.2.	Диаграмма вариантов использования (Use Case)	2			2		Отчет по лабораторной работе
6.3.	Диаграмма классов	2			6		Отчет по лабораторной работе
6.4.	Диаграмма состояний	2			2		Отчет по лабораторной работе
6.5.	Диаграмма деятельности (Activity)	2			2		Отчет по лабораторной работе
6.6.	Диаграмма последовательности	2			2		Отчет по лабораторной работе
6.7.	Диаграмма кооперации	2			2		Отчет по лабораторной работе
6.8.	Диаграммы компонентов и развертывания	2			2		Отчет по лабораторной работе
7.	Тестирование и отладка ПО	5			4	2	
7.1.	Введение в тестирование ПО	1					Коллоквиум по разделам 1-6
7.2.	Модульное тестирование	1				2	учебная дискуссия
7.3.	Функциональное тестирование	1			2		Отчет по лабораторной работе
7.4.	Автоматизация функционального тестирования	1			2		Контрольная работа №2 (по разделам 6-7)
7.5.	Отладка ПО	1					

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Перечень основной литературы

1. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. - Москва: Издательство Юрайт, 2022. - 432 с.
2. Попова, Ю. Б. Тестирование и отладка программного обеспечения: пособие / Ю. Б. Попова. – Минск : БНТУ, 2020. – 66 с.
3. Маран, М. М. Программная инженерия : учебное пособие / М. М. Маран. - Изд. 3-е, стер. - Санкт-Петербург ; Москва ; Краснодар : Лань, 2022. - 194 с. - URL: <https://reader.lanbook.com/book/189470#2>.
4. Фримен, Э. Head First. Паттерны проектирования = Head First. Design Patterns / Эрик Фримен, Элизабет Робсон при участии Кэтти Сьерра и Берта Бейтса ; [пер. с англ. Е. Матвеева]. - Обновленное юбилейное изд. - Санкт-Петербург [и др.]: Питер, 2019. - 651 с. - URL: <https://ibooks.ru/bookshelf/377150>.
5. Паттерны объектно-ориентированного проектирования / Э. Гамма [и др. ; пер. с англ. А. Слинкин]. - Санкт-Петербург ; Москва ; Минск : Питер, 2023. - 446 с. - URL: <https://ibooks.ru/bookshelf/371734>.
6. Чакон, С. Git для профессионального программиста / Чакон С., Штрауб Б.. – СПб.: Питер. Библиотека программиста, 2022. – 496 с.
7. Хориков, В. Принципы юнит-тестирования / Хориков В. - СПб.: Питер. Для профессионалов, 2021. - 320 с. - URL: <https://ibooks.ru/bookshelf/373514>.
8. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения. - СПб.: Питер, 2018. - 352 с. - URL: <https://ibooks.ru/bookshelf/361841>.

Перечень дополнительной литературы

1. Оукс Скотт. Эффективный Java. Тюнинг кода на Java 8, 11 и дальше. — СПб.: Питер, 2021. — 496 с.
2. Маттиас Нобак. Принципы разработки программных пакетов: Проектирование повторно используемых компонентов / пер. с англ. Д. А. Беликова. – М.: ДМК Пресс, 2020. – 274 с.
3. Вигерс К., Битти Д. Разработка требований к программному обеспечению, издание третье. – М: Русская редакция, 2018. – 716 с.
4. Макконнелл С. Совершенный код. Мастер класс / Пер. с англ. — М.: Издательство «Русская редакция», СПб. : Питер, 2016. — 896 с.
5. Бахтизин В.В., Глухова Л.А. Метрология, стандартизация и сертификация в информационных технологиях. Учебное пособие. — В 2-х частях. — Минск: БГУИР, 2016.

6. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. Пер. с англ. – М.: ДМК Пресс, 2014. – 496 с.
7. Ошероув Р. Искусство автономного тестирования с примерами на C#. 2-е издание / пер. с англ. Слинкин А. А. – М.: ДМК Пресс, 2014. – 360 с.
8. Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста. — СПб.: Питер, 2013. — 464 с.
9. Software Engineering компьютерных систем. Парадигмы, технологии и CASE-средства программирования/Е. М. Лаврищева. –К.:Наук. думка, 2013. – 283 с.
10. Липаев В.В. Тестирование компонентов и комплексов программ. Учебник. – М.: СИНТЕГ, 2010. – 400 с.
11. Арлоу Д., Нейштадт И. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование / Пер. с англ. – СПб: Символ-Плюс, 2007. – 624 с.
12. Ларман Крэг. Применение UML и шаблонов проектирования. 2-е издание. Пер. с англ. – М.: Издательский дом "Вильямс", 2007. — 727 с.

Электронные ресурсы

1. Образовательный портал БГУ [Электронный ресурс]. – Режим доступа: <https://edufpmi.bsu.by/course/view.php?id=109> – Дата доступа: 24.11.2022.

Перечень рекомендуемых средств диагностики и методика формирования итоговой отметки

Для диагностики компетенции в рамках учебной дисциплины рекомендуется использовать следующие формы:

1. Устная форма: собеседование, дискуссия.
2. Письменная форма: контрольная работа, коллоквиум, тест.
3. Устно-письменная форма: отчет по лабораторной работе с устной защитой.

В качестве рекомендуемых технических средств используется обучение, организованное на платформе Moodle (<https://edufpmi.bsu.by>).

Формой текущей аттестации по дисциплине учебным планом предусмотрен **зачёт**.

При формировании итоговой отметки используется рейтинговая система оценки знания студента, дающая возможность проследить и оценить динамику процесса достижения целей обучения. Рейтинговая система предусматривает использование весовых коэффициентов для текущего контроля знаний и текущей аттестации студентов по дисциплине.

Примерные весовые коэффициенты, определяющие вклад текущего контроля знаний в итоговую отметку:

- контрольная работа – 30 %;
- выполнение теста – 10%;

- коллоквиум – 10%;
- отчет по лабораторной работе с устной защитой – 50 %.

Итоговая отметка по дисциплине рассчитывается на основе отметки текущей успеваемости и отметки на зачёте с учетом их весовых коэффициентов. Вес отметки по текущей успеваемости составляет 40 %, отметки на зачёте – 60 %.

Примерный перечень заданий для управляемой самостоятельной работы студентов

В качестве заданий для управляемой самостоятельной работы могут быть выданы задания для самостоятельного решения задач по следующим темам:

Тема 6.1. Основные компоненты UML

«CASE-системы проектирования программных средств. Построение канонических UML-диаграмм в программе Modelio-OpenSource» (2 ч.)

Форма контроля – собеседование.

Тема 7.2. Модульное тестирование

«Инструментальные средства модульного тестирования. Построение модульных тестов, используя пакет JUnit.» (2 ч.)

Форма контроля – учебная дискуссия.

Примерная тематика лабораторных занятий

Лабораторная работа 1. Документирование кода.

Лабораторная работа 2. Знакомство с модульным тестированием.

Лабораторная работа 3. Анализ предметной области.

Лабораторная работа 4. Анализ требований к программному продукту (стандарт IEEE 830).

Лабораторная работа 5. Диаграммы UseCase. Сценарии использования программного продукта.

Лабораторная работа 6. Диаграммы деятельности (activity).

Лабораторная работа 7. Логическая модель программного продукта. Диаграммы классов.

Лабораторная работа 8. Физическая модель программного продукта. Кодирование и первичная отладка.

Лабораторная работа 9. Тестирование программного продукта. Опытная эксплуатация, разработка документации.

Рекомендуемая тематика контрольных работ и коллоквиума:

1) Контрольная работа № 1 «Требования к ПО; проектирование архитектуры ПО; принципы дизайна и парадигмы программирования; модели жизненного цикла программного продукта».

2) Контрольная работа № 2 «Unified Modeling Language (UML); тестирование и отладка ПО».

3) Коллоквиум «Канонические диаграммы UML, основные понятия и принципы построения»

С примерным перечнем вопросов и заданий к контрольным работам можно ознакомиться на образовательном портале БГУ (<https://edufpmi.bsu.by/course/view.php?id=109>).

Описание инновационных подходов и методов к преподаванию учебной дисциплины

При организации образовательного процесса используются следующие методы:

- *метод учебной дискуссии*, который предполагает участие студентов в целенаправленном обмене мнениями, идеями для предъявления и/или согласования существующих позиций по определенной проблеме. Использование метода обеспечивает появление нового уровня понимания изучаемой темы, применение знаний (теорий, концепций) при решении проблем, определение способов их решения.

- *метод группового обучения*, который представляет собой форму организации учебно-познавательной деятельности обучающихся, предполагающую функционирование разных типов малых групп, работающих как над общими, так и специфическими учебными заданиями.

В качестве технических средств организации работы в рамках учебной дисциплины рекомендуется использовать Образовательный портал БГУ (<https://edufpmi.bsu.by>) – инструмент с эффективной функциональностью контроля, тренинга и самостоятельной работы.

- *практико-ориентированный подход*, который предполагает освоение содержания образования через решения практических задач; приобретение навыков эффективного выполнения разных видов профессиональной деятельности; ориентацию на генерирование идей, реализацию групповых студенческих проектов; использование процедур, способов оценивания, фиксирующих профессиональные компетенции.

Методические рекомендации по организации самостоятельной работы обучающихся

Для организации самостоятельной работы студентов по учебной дисциплине следует использовать современные информационные ресурсы: разместить на образовательном портале комплекс учебных и учебно-методических материалов (учебно-программные материалы, учебное издание для теоретического изучения дисциплины, методические указания к лабораторным занятиям, материалы текущего контроля и текущей

аттестации, позволяющие определить соответствие учебной деятельности обучающихся требованиям образовательных стандартов высшего образования и учебно-программной документации, в т.ч. вопросы для подготовки к зачету, задания, тесты, вопросы для самоконтроля и др., список рекомендуемой литературы, информационных ресурсов и др.).

При составлении заданий УСР по учебной дисциплине необходимо предусмотреть следующие материалы для выдачи студентам:

1. Порядок работы – даётся пошаговая инструкция по выполнению задания;
2. Теоретический материал – темы для изучения перед выполнением задания;
3. Примеры решения типовых задач по изучаемому теоретическому материалу;
4. Условие и варианты задания – подробное описание условия задания и, опционально, варианты заданий.

С примерными вариантами методических материалов к индивидуальным заданиям и заданиям для УСР можно ознакомиться на образовательном портале БГУ (<https://edufpmi.bsu.by/course/view.php?id=109>)

Примерный перечень вопросов к зачёту

1. Определения требований, классификация требований.
2. Определения требований, потребности.
3. Определения требований, бизнес-требования.
4. Определения требований, пользовательские требования.
5. Определения требований, функциональные требования.
6. Определения требований, бизнес-правила.
7. Определения требований, внешние интерфейсы.
8. Определения требований, атрибуты качества.
9. Определения требований, ограничения.
10. Определения требований, системные требования.
11. Определения требований, features.
12. Инженерия требований к ПО, выявление требований.
13. Инженерия требований к ПО, анализ требований.
14. Инженерия требований к ПО, спецификация требований.
15. Инженерия требований к ПО, валидация требований.
16. Инженерия требований к ПО, верификация требований.
17. Инженерия требований к ПО, управление требованиями.
18. Инженерия требований к ПО, участники процессов (Process Actors).
19. Архитектура ПО, концептуальная архитектура.
20. Архитектура ПО, архитектурный стиль.
21. Архитектура ПО, пилотная архитектура.
22. Архитектура ПО, модуль, компонент, фреймворк.
23. Архитектура ПО, слабая связность.

24. Архитектура ПО, сквозная функциональность.
25. Принципы качественно спроектированной программной архитектуры, разделение функций.
26. Принципы качественно спроектированной программной архитектуры, принцип основной ответственности.
27. Принципы качественно спроектированной программной архитектуры, принцип минимального знания.
28. Принципы качественно спроектированной программной архитектуры, отсутствие повторений.
29. Принципы качественно спроектированной программной архитектуры, проектирование в процессе разработки ПО, проектирование и качество ПО.
30. Принципы качественно спроектированной программной архитектуры, типы программных приложений.
31. Документирование архитектуры ПО, цели документирования.
32. Документирование архитектуры ПО, этапы создания документации.
33. Документирование архитектуры ПО, правила документирования программной архитектуры.
34. Документирование архитектуры ПО, нотации для документирования программной архитектуры.
35. Архитектурные представления и стили, архитектурное представление.
36. Архитектурные представления и стили, архитектурная модель 4+1.
37. Архитектурные представления и стили, подход «Views and Beyond».
38. Архитектурные представления и стили, модульные представления.
39. Архитектурные представления и стили, наиболее распространенные модульные представления.
40. Архитектурные представления и стили, компонентные представления.
41. Архитектурные представления и стили, представления развертывания.
42. Архитектурные представления и стили, микроархитектура, внедрение зависимости.
43. Архитектурные представления и стили, схемы данных.
44. Архитектурные представления и стили, компонентные архитектурные стили.
45. Архитектурные представления и стили, сочетание архитектурных стилей.
46. Архитектурные представления и стили, интеграция приложений.
47. Парадигмы программирования, ограничения, связанные с парадигмой программирования.
48. Парадигмы программирования, структурное программирование.
49. Парадигмы программирования, объектно-ориентированное программирование.
50. Парадигмы программирования, функциональное программирование.
51. Принципы дизайна SOLID, принцип единственной ответственности.
52. Принципы дизайна SOLID, принцип открытости / закрытости.
53. Принципы дизайна SOLID, принцип подстановки Барбары Лисков.

54. Принципы дизайна SOLID, принцип разделения интерфейсов.
55. Принципы дизайна SOLID, принцип инверсии зависимости.
56. Модели жизненного цикла ПП, каскадная модель, основные принципы каскадной модели.
57. Модели жизненного цикла ПП, преимущества каскадной модели.
58. Модели жизненного цикла ПП, недостатки каскадной модели.
59. Модели жизненного цикла ПП, усовершенствованная каскадная модель.
60. Модели жизненного цикла ПП, спиральная модель.
61. Модели жизненного цикла ПП, преимущества спиральной модели.
62. Модели жизненного цикла ПП, MBASE принципы.
63. Модели жизненного цикла ПП, эволюция спиральной модели.
64. Модели жизненного цикла ПП, недостатки спиральной модели.
65. Модели жизненного цикла ПП, V-образная модель.
66. Модели жизненного цикла ПП, инкрементная (пошаговая) модель.
67. Модели жизненного цикла ПП, модель быстрого прототипирования.
68. Модели промышленных технологий создания ПП, модель Microsoft Solution Framework (MSF).
69. Модели промышленных технологий создания ПП, Agile-методологии разработки.
70. Модели промышленных технологий создания ПП, модель MSF for Agile Software Development.
71. Модели промышленных технологий создания ПП, модель Rational Unified Process (RUP).
72. Модели промышленных технологий создания ПП, модель Extreme Programming (XP).
73. Основные компоненты UML, структура UML.
74. Основные компоненты UML, основные пакеты метамодели UML.
75. Основные компоненты UML, описание метамодели UML.
76. Основные компоненты UML, диаграммы UML, особенности изображения диаграмм UML.
77. Диаграмма вариантов использования (Use Case), вариант использования.
78. Диаграмма вариантов использования (Use Case), актеры.
79. Диаграмма вариантов использования (Use Case), интерфейсы.
80. Диаграмма вариантов использования (Use Case), примечания.
81. Диаграмма вариантов использования (Use Case), отношения на диаграмме вариантов использования.
82. Диаграмма классов, элементы диаграммы классов.
83. Диаграмма классов, класс, имя класса, атрибуты класса, операции.
84. Диаграмма классов, отношения между классами.
85. Диаграмма классов, интерфейсы.
86. Диаграмма классов, объекты.
87. Диаграмма классов, шаблоны или параметризованные классы.
88. Диаграмма состояний, автоматы.

89. Диаграмма состояний, состояние, имя состояния, список внутренних действий, начальное состояние, конечное состояние, переход, событие.
90. Диаграмма состояний, сторожевое условие, выражение действия.
91. Диаграмма состояний, составное состояние и подсостояние, последовательные подсостояния, параллельные подсостояния, историческое состояние.
92. Диаграмма состояний, сложные переходы, переходы между параллельными состояниями, переходы между составными состояниями.
93. Диаграмма состояний, синхронизирующие состояния. Построение диаграмм состояний.
94. Диаграмма деятельности, состояние действия.
95. Диаграмма деятельности, переходы, дорожки.
96. Диаграмма деятельности, объекты, построение диаграмм деятельности.
97. Диаграмма последовательности, объекты, линия жизни объекта.
98. Диаграмма последовательности, фокус управления, сообщения.
99. Диаграмма последовательности, ветвление потока управления.
100. Диаграмма последовательности, временные ограничения, комментарии или примечания.
101. Диаграмма последовательности, построение диаграмм последовательности.
102. Диаграмма кооперации, кооперация, диаграмма кооперации уровня спецификации.
103. Диаграмма кооперации, объекты, мультиобъект, активный объект, составной объект.
104. Диаграмма кооперации, связи, стереотипы связей.
105. Диаграмма кооперации, сообщения, формат записи сообщений.
106. Диаграмма кооперации, построение диаграмм кооперации.
107. Диаграммы компонентов и развертывания, компоненты, имя компонента, виды компонентов.
108. Диаграммы компонентов и развертывания, интерфейсы, зависимости, построение диаграмм компонентов.
109. Диаграммы компонентов и развертывания, диаграмма развертывания, узел, соединения, построение диаграмм развертывания.
110. Введение в тестирование ПО, виды тестирования ПО.
111. Введение в тестирование ПО, экономическая сторона тестирования.
112. Модульное тестирование, модульное тестирование и его задачи.
113. Модульное тестирование, обзоры программного кода.
114. Модульное тестирование, тестирования структуры модулей.
115. Модульное тестирование, тестирование взаимодействия модулей.
116. Модульное тестирование, стратегии пошагового тестирования.
117. Модульное тестирование, объектно-ориентированное тестирование.
118. Модульное тестирование, автоматизация модульного тестирования.
119. Модульное тестирование, семейство xUnit.

120. Модульное тестирование, универсальные средства автоматизации тестирования.
121. Функциональное тестирование, планирование функционального тестирования.
122. Функциональное тестирование, тестовый план, разработка тестовых случаев, выбор тестовых данных.
123. Функциональное тестирование, ошибка, свойства ошибки, составления отчетов об ошибках.
124. Функциональное тестирование, системы документирования/отслеживания ошибок, жизненный цикл ошибки.
125. Функциональное тестирование, реализация тестов: приемочный тест, позитивный тест, негативный тест.
126. Функциональное тестирование, особенности функционального тестирования.
127. Автоматизация функционального тестирования, плюсы/минусы автоматизации тестирования.
128. Автоматизация функционального тестирования, требования к автоматизированным тестам.
129. Автоматизация функционального тестирования, методы автоматизации функционального тестирования.
130. Автоматизация функционального тестирования, метод Play&Record.
131. Автоматизация функционального тестирования, метод функциональной декомпозиции.
132. Автоматизация функционального тестирования, метод Data-driven.
133. Автоматизация функционального тестирования, метод Keyword-driven.
134. Автоматизация функционального тестирования, семейство Selenium и его возможности.
135. Автоматизация функционального тестирования, проблемы внедрения автоматизации тестирования.
136. Отладка ПО, методика отладки.
137. Отладка ПО, методы отладки.
138. Отладка ПО, средства отладки.

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)
Модели данных и СУБД	Информационных систем управления	Нет	Изменений не требуется (протокол № 6 от 24.11.2022 г.).

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**
на ____ / ____ учебный год

№№ ПП	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры информационных систем управления (протокол № ____ от _____ 200_ г.)

Заведующий кафедрой

(степень, звание)

(подпись)

(И.О.Фамилия)

УТВЕРЖДАЮ
Декан факультета

(степень, звание)

(подпись)

(И.О.Фамилия)