

**Государственное образовательное учреждение  
«Приднестровский государственный университет им. Т.Г. Шевченко»**

**Инженерно-технический институт**

**Кафедра «Программное обеспечение вычислительной техники  
и автоматизированных систем»**

УТВЕРЖДАЮ

Директор института, доцент

 Ф.Ю. Бурменко

«14» 09 2018 г.



# **РАБОЧАЯ ПРОГРАММА**

на 2018/2019 учебный год

учебной дисциплины

## **Б1.В.ОД.8 «ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ»**

Направление подготовки:  
**09.03.04 Программная инженерия**

Профиль подготовки  
**Разработка программно-информационных систем**

Для набора  
**2016 года**

Квалификация (степень) выпускника  
**бакалавр**

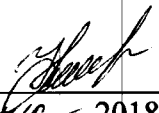
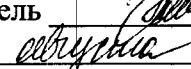
Форма обучения:  
**очная**

Тирасполь, 2018

Рабочая программа дисциплины «Проектирование и архитектура программных систем» / сост. О.С. Белоконь – Тирасполь: ГОУ ПГУ, 2018. – 18 с.

Рабочая программа предназначена для преподавания обязательной дисциплины вариативной части дисциплин (модулей) студентам очной формы обучения по направлению подготовки 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

Рабочая программа составлена с учетом Федерального Государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ», утвержденного приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. № 229.

Составитель  /О.С. Белоконь, ст. преподаватель кафедры ПОВТиАС  
«»  2018 г.

## 1 Цели и задачи освоения дисциплины

*Цель дисциплины:* изучение теоретических основ и приобретение практических навыков в области проектирования и разработки структуры и архитектуры программных систем.

Предметом изучения дисциплины «Проектирование и архитектура программных систем» являются модели и структуры программных систем, методы и этапы проектирования программных систем, особенности различных архитектур, современные подходы к построению программных систем и используемые в этом процессе программные продукты.

Задачи изучения дисциплины:

- подготовка студентов в области проектирования программных систем;
- формирование подходов к выполнению самостоятельных исследований студентами в области проектирования архитектуры программных систем.

## 2 Место дисциплины в структуре ООП ВО

Б1.В.ОД.8. Дисциплины (модули). Вариативная часть. Обязательные дисциплины.

Общая трудоёмкость дисциплины составляет 3 зачетные единицы, 108 часов.

Изучение данной дисциплины базируется на следующих дисциплинах: информатика, объектно-ориентированное программирование, введение в программную инженерию. Основные положения дисциплины могут быть использованы в практической деятельности специалиста, а также при изучении дисциплин: конструирование программного обеспечения, тестирование и надежность программного обеспечения.

Для успешного освоения дисциплины необходимо:

- понимать основные определения и понятия теории проектирования программных систем, правила проектирования программных систем и проектирования архитектур программных систем;
- владеть основными методами, способами разработки нормативных документов на этапе проектирования программных систем;
- иметь навыки работы с системами автоматического проектирования архитектуры программных систем;
- уметь разрабатывать диаграммы на языке UML.

## 3 Требования к результатам освоения дисциплины

Изучение дисциплины направлено на формирование следующих компетенций: ОПК-3, ПК-7, ПК-16. Расшифровка компетенций дана в следующей таблице.

Таблица 1 – Формулировка компетенции для направления 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ»

| Код компетенции | Формулировка компетенций                                                                                                                                             |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ОПК-3           | готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов                                   |
| ПК-7            | владением методами управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения |
| ПК-16           | способностью формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта                                  |

В результате освоения дисциплины обучающиеся должны:

### 3.1. Знать:

- базовые принципы разработки программного обеспечения;
- основные принципы построения программных систем с использованием различных сред для построения промежуточного программного обеспечения;

- принципы модификации проектов с использованием различных подходов к управлению изменениями;
- концепции и стратегии архитектурного проектирования программного продукта;
- методы и процессы разработки требований к программным продуктам;
- способы оценки рисков проектирования, конструирования программного продукта.

### 3.2. Уметь:

- применять множество различных шаблонов проектирования, сред разработки и архитектур в проектировании разнообразных программных систем;
- проектировать компоненты программного продукта;
- применять основы информатики и программирования к проектированию программных продуктов;
- использовать различные технологии разработки программного обеспечения;
- оформлять методические материалы и пособия по применению программных систем;
- понимать методы управления процессами разработки требований, оценки рисков, приобретения, проектирования, конструирования, тестирования, эволюции и сопровождения;
- разрабатывать и специфицировать требования для компонентов программного продукта;
- самостоятельно осваивать современные инструментальные средства;
- организовывать и осуществлять процессы проектирования программных средств;
- самостоятельно осваивать передовые технологии разработки программного обеспечения; разрабатывать спецификации программного обеспечения.

### 3.3. Владеть:

- технологиями и инструментами разработки программного обеспечения;
- методами и средствами разработки спецификаций для компонентов программного продукта;
- навыками проектирования и применения современных инструментальных средств разработки программного обеспечения;
- языками спецификаций, языками программирования.

Рабочая программа учебной дисциплины рассчитана на 52 часа аудиторных занятий, в том числе 18 часов отводится на лекционные занятия, 34 часа – на лабораторные занятия.

С целью систематизации и закрепления полученных теоретических знаний и практических умений в рабочей программе учебной дисциплины предусмотрена самостоятельная работа студентов. Внеаудиторная самостоятельная работа студентов – 56 часов.

Для проверки знаний студентов в рабочей программе указаны по окончании изучения, каких разделов следует проводить рубежный контроль. Учебная дисциплина изучается один семестр и заканчивается промежуточной аттестацией студентов в форме зачета.

## 4. Структура и содержание дисциплины

### 4.1 Распределение трудоёмкости в часах по видам аудиторной и самостоятельной работы студента

| Семестр | Количество часов          |             |        |                         |                       |                | Форма<br>итог.<br>контроля |
|---------|---------------------------|-------------|--------|-------------------------|-----------------------|----------------|----------------------------|
|         | Трудоемкость<br>з.е./часы | В том числе |        |                         |                       | Сам.<br>работа |                            |
|         |                           | Аудиторных  |        |                         |                       |                |                            |
|         |                           | Всего       | Лекций | Практических<br>занятия | Лабораторных<br>работ |                |                            |
| 6       | 3/108                     | 52          | 18     | -                       | 34                    | 56             | Зачёт                      |
| Итого   | 3/108                     | 52          | 18     | -                       | 34                    | 56             | Зачёт                      |

#### 4.2 Распределение видов учебной работы и их трудоёмкости по разделам дисциплины

| № раздела                  | Наименование раздела дисциплины (модуля)              | Количество часов |                   |          |           |                           |
|----------------------------|-------------------------------------------------------|------------------|-------------------|----------|-----------|---------------------------|
|                            |                                                       | Всего            | Аудиторная работа |          |           | Внеаудиторная работа (СР) |
|                            |                                                       |                  | ЛК                | ПЗ       | ЛР        |                           |
| 1                          | Архитектурное проектирование                          | 16               | 2                 | -        | 6         | 8                         |
| 2                          | Архитектура распределенных систем                     | 18               | 4                 | -        | 6         | 8                         |
| 3                          | Объектно-ориентированное проектирование               | 18               | 2                 | -        | 6         | 10                        |
| 4                          | Проектирование систем реального времени               | 20               | 4                 | -        | 6         | 10                        |
| 5                          | Проектирование с повторным использованием компонентов | 18               | 2                 | -        | 6         | 10                        |
| 6                          | Проектирование интерфейса пользователя                | 18               | 4                 | -        | 4         | 10                        |
| <b>Итого по дисциплине</b> |                                                       | <b>3/108</b>     | <b>18</b>         | <b>-</b> | <b>34</b> | <b>56</b>                 |

#### 4.3 Тематический план по видам учебной деятельности

##### Лекции

| № п/п        | Номер раздела дисциплины | Объем часов | Тема лекции                                                                                                                                                     | Учебно-наглядные пособия |
|--------------|--------------------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| 1            | 2                        | 3           | 4                                                                                                                                                               | 5                        |
| 1            | 1                        | 2           | <b>Архитектурное проектирование:</b> Структурирование системы. Модели управления. Модульная декомпозиция. Проблемно-зависимые архитектуры                       | слайды презентации       |
| 2            | 2                        | 2           | <b>Архитектура распределенных систем:</b> Много-процессорная архитектура. Архитектура клиент/сервер.                                                            | слайды презентации       |
| 3            | 2                        | 2           | <b>Архитектура распределенных систем:</b> Архитектура распределённых объектов.                                                                                  | слайды презентации       |
| 4            | 3                        | 2           | <b>Объектно-ориентированное проектирование:</b> Объекты и классы объектов. Процесс объектно-ориентированного проектирования. Модификация системной архитектуры. | слайды презентации       |
| 5            | 4                        | 2           | <b>Проектирование систем реального времени:</b> Проектирование систем. Управляющие программы.                                                                   | слайды презентации       |
| 6            | 4                        | 2           | <b>Проектирование систем реального времени:</b> Системы наблюдения и управления. Системы сбора данных.                                                          | слайды презентации       |
| 7            | 5                        | 2           | <b>Проектирование с повторным использованием компонентов:</b> Покомпонентная разработка. Семейства приложений. Проектные паттерны.                              | слайды презентации       |
| 8            | 6                        | 2           | <b>Проектирование интерфейса пользователя:</b> Принципы проектирования интерфейса пользователя. Взаимодействие с пользователем..                                | слайды презентации       |
| 9            | 6                        | 2           | <b>Проектирование интерфейса пользователя:</b> Средства поддержки пользователя. Оценивание интерфейса                                                           | слайды презентации       |
| <b>Итого</b> |                          | <b>18</b>   |                                                                                                                                                                 |                          |

**Практические (семинарные) занятия – учебным планом не предусмотрено**

**Лабораторные работы**

| <b>№ п/п</b> | <b>Номер раздела дисциплины</b> | <b>Объем часов</b> | <b>Тема лабораторного занятия</b>                                                                                                                      | <b>Учебно-наглядные пособия</b> |
|--------------|---------------------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|
| 1            | 1                               | 2                  | Разработка технического задания: основные сведения.                                                                                                    | электр. вариант ЛБ              |
| 2            | 1                               | 2                  | Разработка технического задания: основные сведения                                                                                                     | электр. вариант ЛБ              |
| 3            | 1                               | 2                  | Разработка технического задания: основные сведения                                                                                                     | электр. вариант ЛБ              |
| 4            | 2                               | 2                  | Применение структурного подхода в анализе требований и определении спецификаций программных систем: основные сведения; диаграммы переходов состояний   | электр. вариант ЛБ              |
| 5            | 2                               | 2                  | Применение структурного подхода в анализе требований и определении спецификаций программных систем: функциональные диаграммы; диаграммы потоков данных | электр. вариант ЛБ              |
| 6            | 2                               | 2                  | Применение структурного подхода в анализе требований и определении спецификаций программных систем: диаграммы «сущность-связь»                         | электр. вариант ЛБ              |
| 7            | 3                               | 2                  | Применение объектно-ориентированного подхода в анализе и проектировании программных систем: диаграммы вариантов использования                          | электр. вариант ЛБ              |
| 8            | 3                               | 2                  | Применение объектно-ориентированного подхода в анализе и проектировании программных систем: диаграммы деятельности и последовательности                | электр. вариант ЛБ              |
| 9            | 3                               | 2                  | Применение объектно-ориентированного подхода в анализе и проектировании программных систем: диаграммы классов.                                         | электр. вариант ЛБ              |
| 10           | 4                               | 2                  | Проектирование программных систем при структурном подходе: структурная схема                                                                           | электр. вариант ЛБ              |
| 11           | 4                               | 2                  | Проектирование программных систем при структурном подходе: Структурная схема                                                                           | электр. вариант ЛБ              |
| 12           | 4                               | 2                  | Проектирование программных систем при структурном подходе: функциональная схема.                                                                       | электр. вариант ЛБ              |
| 13           | 5                               | 2                  | Разработка прототипа программных систем: основные сведения о прототипах.                                                                               | электр. вариант ЛБ              |
| 14           | 5                               | 2                  | Разработка прототипа программных систем: виды прототипов                                                                                               | электр. вариант ЛБ              |
| 15           | 5                               | 2                  | Разработка прототипа программных систем: построение прототипа.                                                                                         | электр. вариант ЛБ              |
| 16           | 6                               | 2                  | Проектирование интерфейса пользователя: основные правила создания; принципы разработки                                                                 | электр. вариант ЛБ              |
| 17           | 6                               | 2                  | Проектирование интерфейса пользователя: взаимодействие между пользователем и ПК; размещение информации на экране.                                      | электр. вариант ЛБ              |
|              | <b>Итого</b>                    | <b>34</b>          |                                                                                                                                                        |                                 |

# Самостоятельная работа студента

| Раздел дисциплины | № п/п | Тема и вид СРС                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Трудоемкость (в часах) |
|-------------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| 1                 | 2     | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 4                      |
| Раздел 1          | 1     | <p>Тема: Варианты архитектур программных систем. Архитектура, основанная на уровнях абстракций. Архитектуры, основанные на портах. Архитектуры независимых компонентов. Архитектуры, основанные на потоках данных.</p> <p>СРС №1: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации</p> <p>Литература:</p> <ol style="list-style-type: none"> <li>1. Басс Л., Клементс П., Кацман Р. Архитектура программного обеспечения на практике. 2-е издание. – СПб.: Питер, 2006</li> <li>2. Майерс Г. Надежность программного обеспечения: Пер. с англ. – М.: Мир, 1980.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 8                      |
| Раздел 2          | 2     | <p>Тема: Методы проектирования компонентных и модульных архитектур программных систем. Рефакторинг архитектуры программных систем. Формальное описание методики разработки модульной архитектуры программных систем. Модульность и ее характеристики. Оценка сложности модульных иерархических структур.</p> <p>СРС №2: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации</p> <p>Литература:</p> <ol style="list-style-type: none"> <li>1. Полис Г., Огастин Л., Мадхар Д. Разработка программных проектов: на основе Rational Unified Process (RUP). – М.: ООО «Бином-Пресс», 2009</li> <li>2. Майерс Г. Надежность программного обеспечения: Пер. с англ. – М.: Мир, 1980.</li> <li>3. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.: Питер, 2002.</li> <li>4. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной. – М.: ИД «Форум»: Инфра-М, 2008</li> <li>5. Ксензов М. Рефакторинг архитектуры программного обеспечения: выделение слоев. Труды ИПИРАН</li> </ol> | 8                      |
| Раздел 3          | 3     | <p>Тема: Метод функционального моделирования. Функциональные диаграммы. Диаграммы потоков данных. Диаграммы переходов состояний. Анализ требований и определение спецификаций при объектном подходе к проектированию.</p> <p>СРС №3: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации</p> <p>Литература:</p> <ol style="list-style-type: none"> <li>1. Майерс Г. Надежность программного обеспечения: Пер. с англ. – М.: Мир, 1980.</li> <li>2. Технологии разработки программного обеспечения: Учебник/ С. Орлов. — СПб.: Питер, 2002.</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 10                     |

|              |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |
|--------------|---|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
|              |   | 3. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной. – М.: ИД «Форум»: Инфра-М, 2008                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |
| Раздел 4     | 4 | Тема: Модель проектирования ПС как последовательная трансляция требований, предъявляемых к системе. Методология решения задач проектирования по Г. Майерсу. Уровни требований к программным системам.<br>СРС №4: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации<br>Литература:<br>1. Майерс Г. Надежность программного обеспечения: Пер. с англ. – М.: Мир, 1980.<br>2. Э.Халл, К. Джексон: Разработка и управление требованиями<br>3. Колин Ю. Применение Rational Software Architect в разработке, управляемой моделями и на основе шаблонов: Часть 2. Инструментальная поддержка разработки.<br><a href="http://www.ibm.com/developerworks/ru/library/0116_yu/#author1">http://www.ibm.com/developerworks/ru/library/0116_yu/#author1</a> | 10        |
| Раздел 5     | 5 | Тема: Промежуточное программное обеспечение. Стандартное программное обеспечение для организации промежуточного слоя. CORBA, Enterprise java Beans, ASP.Net, JMS, OMG, ORB, RMI, SOAP, WSDL.<br>СРС №5: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации<br>Литература:<br>1. Цимбал А. Технология CORBA для профессионалов – С-Пб. Питер, 2001.<br>2. Р. Дж. Оберг Технология COM+. Основы и программирование – М.: Вильямс, 2000                                                                                                                                                                                                                                                                                                             | 10        |
| Раздел 6     | 6 | Тема: Предотвращение, обнаружение и исправление ошибок интерфейса пользователя. Общие требования к графическому интерфейсу пользователя.<br>СРС №6: изучение материалов дополнительной литературы, поиск и анализ материала в электронных источниках информации<br>Литература:<br>1. Джеф Раскин, «Интерфейс: новые направления в проектировании компьютерных систем»<br>2. Алан Купер, «Об интерфейсе. Основы проектирования взаимодействия»                                                                                                                                                                                                                                                                                                                                                             | 10        |
| <b>Итого</b> |   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | <b>56</b> |

5. Примерная тематика курсовых проектов (работ) – учебным планом не предусмотрено.

#### 6. Образовательные технологии

| Семестр | Вид занятия (Л, ПР, ЛР) | Используемые интерактивные образовательные технологии                                                 | Количество часов |
|---------|-------------------------|-------------------------------------------------------------------------------------------------------|------------------|
| 6       | Л                       | – информационно-развивающие технологии;<br>– компьютерные технологии обучения<br>– проблемная лекция, | 18               |



|  |    |                                                                                                                                                                                                                          |           |
|--|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|
|  |    | <ul style="list-style-type: none"> <li>– лекция-дискуссия (лекция-обсуждение),</li> <li>– лекция с заранее запланированным</li> <li>– ошибками (метод контрольного изложения),</li> <li>– лекция-конференция;</li> </ul> |           |
|  | ЛР | компьютерные технологии обучения<br>технология учебного проектирования<br>работа в малых группах                                                                                                                         | 34        |
|  |    | <b>Итого</b>                                                                                                                                                                                                             | <b>52</b> |

**7. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов:**

Итоговой формой контроля знаний, умений и навыков по дисциплине является зачет.

Для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов:

*Вопросы к модульному контролю № 1*

1. Почему архитектуру системы необходимо разработать до окончания создания спецификации.
2. Перечислите преимущества и недостатки различных структурных моделей.
3. Предложите подходящую структурную модель для перечисленных ниже систем. Обоснуйте свой выбор.

- Система автоматической продажи железнодорожных билетов.
- Система видеоконференций, управляемая компьютером, с возможностью одновременного просмотра компьютерных, аудио- и видеоданных несколькими участниками.
- Робот-уборщик, который убирает относительно свободные пространства, например коридоры. Робот должен фиксировать стены и другие преграды.

4. На основе выбранной модели разработайте архитектуру для выше описанных систем. Сделайте предположения о системных требованиях.

5. Объясните, почему модель управления вызова-возврата обычно не подходит для систем реального времени, управляющих определенным процессом.

6. Предложите подходящую модель управления для перечисленных ниже систем. Обоснуйте свой выбор.

- Пакетная система обработки данных, которая на основании информации об отработанных часах и ставок заработной платы формирует заявки на зарплату персонала и передает информацию в банк.
- Набор инструментальных программных средств от разных производителей, которые должны работать совместно.
- Телевизионный контроллер, который отвечает на сигналы, поступающие от удаленного блока управления.

7. Назовите преимущества и недостатки модели потоков данных и объектной модели в предположении, что необходимо разработать как локальную, так и распределенную версии программного приложения.

8. Объясните, почему распределенные системы всегда более масштабируемы, чем централизованные. Какой вероятный предел масштабируемости программных систем?

9. В чем основное отличие между моделями толстого и тонкого клиента в разработке систем клиент/сервер?

10. На основе модели приложения, изображенной на рисунке, рассмотрите возможные



проблемы, которые могут возникнуть при преобразовании системы 1980-х годов, реализованной на мейнфрейме и предназначенной для работы в сфере здравоохранения, в систему архитектуры клиент/сервер.

11. Объясните, почему использование распределенных объектов совместно с брокером запросов к объектам упрощает реализацию масштабируемых систем клиент/сервер. Проиллюстрируйте свой ответ примером.

12. Каким образом используется язык IDL для поддержки взаимодействия между объектами, реализованными на разных языках программирования? Объясните, почему такой подход может вызвать проблемы, связанные с производительностью, если между языками, которые используются при реализации объектов, имеются радикальные различия.

13. Какие базовые средства должен предоставлять брокер запросов к объектам?

14. Можно показать, что разработка стандартов CORBA для горизонтальных и вертикальных компонентов ограничивает конкуренцию. Если они уже созданы и адаптированы, это препятствует разработке лучших компонентов более мелкими компаниями. Обсудите роль стандартизации в поддержке или ограничении конкуренции на рынке программного обеспечения.

15. Объясните, почему в проектировании систем применение подхода, который полагается на слабо связанные объекты, скрывающие информацию о своем представлении, приводит к созданию системной архитектуры, которую затем можно легко модифицировать.

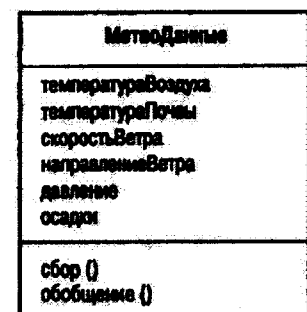
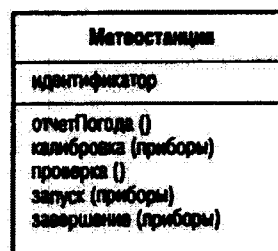
16. Покажите на примерах разницу между объектом и классом объектов.

17. При каких условиях можно разрабатывать систему, в которой объекты выполняются параллельно?

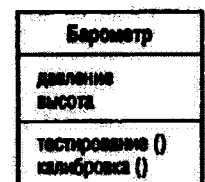
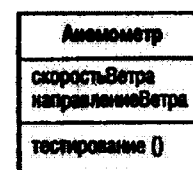
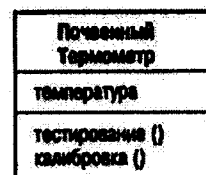
18. С помощью графической системы нотации UML спроектируйте следующие классы объектов с определенными атрибутами и операциями:

- телефон;
- принтер персонального компьютера;
- персональная стереосистема;
- банковские расчеты;
- каталог библиотеки.

19. Разработайте более детальный проект метеостанции, добавив описания интерфейсов объектов, изображенных на рисунке. Они могут быть записаны с помощью языков Java, C++ или UML.



20. Разработайте проект метеостанции, показывающий взаимодействие между подсистемой сбора данных и приборами, собирающими данные. Воспользуйтесь диаграммой последовательностей.



21. Определите возможные объекты в следующих системах, применяя при этом объектно-ориентированный подход.

- Система "Дневник группы" поддерживает расписание собраний и ветрен в группе сотрудников. Для организации встречи, в которой участвует группа людей, система находит общие для всех личных дневников свободные "окна" и назначает эту встречу на определен-

ное время. Если система не находит общих "окон", то начинает взаимодействовать с пользователями, чтобы реорганизовать личные дневники и тем самым создать "окно" для встречи.

- Установлена полностью автоматизированная бензоколонка. Водитель вставляет кредитную карточку в считывающее устройство, связанное с насосом; карточка по линиям коммуникаций проверяется кредитной компанией, устанавливается требуемое количество бензина. Затем автомобиль заправляется горючим. Когда подача прекращается, с кредитной карточки водителя снимается стоимость полученного бензина. Кредитная карточка возвращается после вычета водителю. Если карточка неверна, она возвращается водителю перед подачей топлива.

22. Запишите точные определения интерфейсов на языке Java или C++ для объектов, определенных в предыдущем вопросе.

23. Нарисуйте диаграмму последовательностей, в которой отображены взаимодействия между объектами в системе "Дневник группы".

24. Нарисуйте диаграмму состояний, на которой отображены возможные изменения состояний для одного или более объектов, определенных в задании 21.

### *Вопросы к модульному контролю № 2*

1. Почему системы реального времени обычно реализованы как множество параллельных процессов? Проиллюстрируйте свой ответ примерами.

2. Объясните, почему объектно-ориентированные методы разработки ПО не всегда подходят к системам реального времени.

3. Нарисуйте диаграммы состояний управляющего ПО для следующих систем.

- Автоматическая стиральная машина с разными программами для разных типов белья.

- Программное обеспечение для проигрывателя компакт-дисков.

- Телефонный автоответчик, который записывает входящие сообщения и отображает количество полученных сообщений на жидкокристаллическом экране. Система должна определить телефон звонившего, вывести на экран последовательность чисел (идентифицированных как тоновый набор) и хранить записанные сообщения, которые затем можно прослушать.

- Автомат по выдаче напитков, который может налить кофе с молоком и сахаром или без них. Пользователь бросает монету и с помощью нажатия кнопок на автомате выбирает нужный режим. Автомат выдает чашку с растворимым кофе. Пользователь затем ставит чашку под кран, нажимает другую кнопку и автомат наливает в чашку горячую воду.

4. Используя методы проектирования систем реального времени спроектируйте заново систему сбора данных от метеостанций, рассмотренную в разделе 3, в виде системы "стимул-ответ".

5. Спроектируйте архитектуру процессов для системы наблюдения, собирающей данные с группы датчиков, измеряющих состав воздуха и расположенных вокруг города. В системе 5000 датчиков, организованных в группы по 100 штук. Каждый датчик должен проверяться 4 раза в секунду. Если более 30% датчиков в группе зафиксируют, что качество воздуха ниже допустимого уровня, активизируется предупреждающий световой сигнал, все датчики передают собранные данные центральному компьютеру, который каждые 15 мин генерирует отчет о составе воздуха в городе.

6. Обсудите сильные и слабые стороны Java как языка программирования для реализации систем реального времени.

7. Система безопасности поезда автоматически закрывает двери, если скорость поезда превышает предельную для данного участка трассы или если при выходе на участок пути горит красный свет (т.е. въезд на участок запрещен). Идентифицируйте входные сигналы, которые должна обрабатывать бортовая система управления поездом, и связанные с ними ответные сигналы.

8. Предположите вероятную архитектуру процессов такой системы.

9. Если в бортовой системе безопасности поезда при сборе данных с путевых передатчиков используются периодические процессы, какую частоту сбора данных следует запланировать, чтобы система гарантированно получала информацию от передатчиков? Обоснуйте свой ответ.

10. Каковы основные технические и нетехнические факторы, затрудняющие повторное использование программного обеспечения?

11. Объясните, почему сокращение расходов при повторном использовании компонентов не прямо пропорционально размерам повторно используемых компонентов.

12. Приведите четыре аргумента против повторного использования компонентов.

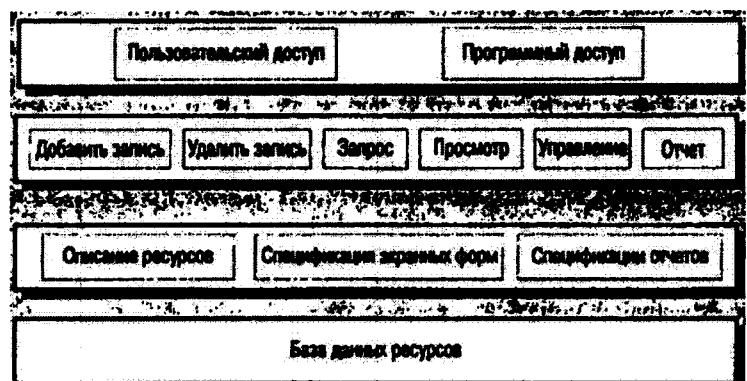
13. Предположите возможные интерфейсы запросов и поставщиков сервисов для следующих компонентов.

- Компонент, реализующий счет в банке.
- Компонент, реализующий не зависящую от языка клавиатуру. Клавиатуры в разных странах имеют различную организацию клавиш и разные наборы символов.

14. Чем отличается повторное использование объектной структуры приложения от повторного использования коммерческих продуктов? Почему иногда проще повторно использовать коммерческий продукт, чем объектную структуру приложения?

15. На примере метеорологической станции, описанной в разделе 3, предложите архитектуру семейства приложений, которые связаны с удаленным наблюдением и сбором метеоданных.

16. На примере семейства приложений по управлению ресурсами (см. рисунок) подумайте, какие методы необходимо добавить или изменить, чтобы можно было делать повторный заказ на отдельные виды ресурсов, если их количество становится меньше некоторой заданной величины.



17. Почему паттерны — эффективный способ повторного использования в проектировании? Каковы недостатки этого подхода?

18. Повторное использование увеличивает количество вопросов о собственности, охраняемой авторским и интеллектуальным правом. Если заказчик оплачивает разработчику ПО заказ на разработку какой-либо системы, кто имеет право повторно использовать созданный код? Имеет ли право разработчик использовать этот код в качестве основы для базового компонента? Какими должны быть механизмы оплаты труда разработчика повторно используемых компонентов? Обсудите эти и другие этические вопросы, связанные с повторным использованием программного обеспечения.

19. В разделе отмечалось, что объекты, которыми манипулирует пользователь, должны отображать его понятия предметной области приложения ПО (а не компьютерной предметной

области). Предложите подходящие объекты манипулирования для следующих типов пользователей и систем.

- Автоматизированный каталог товаров для ассистента на складе.
- Система наблюдения за безопасностью самолета для летчика гражданской авиации.
- Финансовая база данных для менеджера.
- Система управления патрульными машинами для полицейского.

20. Опишите ситуации, в которых неразумно или невозможно поддерживать интерфейс пользователя.

21. Какие факторы следует учитывать при проектировании интерфейсов, использующих меню, для таких систем, как банкоматы? Опишите основные черты интерфейса банкомата, которым вы пользуетесь.

22. Предложите способы адаптации пользовательского интерфейса в системах электронной коммерции (например, виртуального книжного магазина или магазина музыкальных дисков) для пользователей, имеющих физические недостатки, например плохое зрение или проблемы опорно-двигательной системы.

23. Обсудите преимущества графического способа отображения информации и приведите четыре примера приложений, в которых более уместно использовать графическое представление числовых данных, а не табличное.

24. Какими основными принципами следует руководствоваться при использовании цветов в интерфейсах пользователя? Предложите более эффективный способ использования цветов в интерфейсе любого известного вам приложения.

25. Рассмотрите сообщения об ошибках, генерируемые операционными системами MS Windows, Unix, MacOS или какой-либо другой. Как их можно улучшить?

26. Составьте анкету по сбору данных об интерфейсе какой-либо известной вам программы (например, текстового редактора). Если есть возможность, распространите эту анкету среди других пользователей и попытайтесь оценить результаты анкетирования. Что вы узнали об интерфейсе программы из анкет?

27. Обсудите, этично ли разрабатывать программные системы, не согласовав с конечными пользователями те элементы системы, которые они будут контролировать.

28. С какими этическими проблемами сталкиваются разработчики интерфейсов, когда пытаются согласовать запросы конечных пользователей системы с требованиями организации, которая оплачивает разработку данной системы?

#### *Вопросы к защите лабораторных работ*

1. Перечислите основные этапы разработки программных систем.
2. Что включает в себя предпроектные исследования?
3. Назовите основные разделы технического задания.
4. Перечислите функциональные и эксплуатационные требования к программной системе. В чем их принципиальное различие?
5. Приведите основные разделы документа «Техническое задание».
6. Перечислите принципы создания диаграмм. Объясните назначение каждого вида диаграммы.
7. В чем заключаются основные принципы структурного подхода?
8. Что общего и в чем различия между функциональной моделью SADT и диаграммой потоков данных?
9. Назовите достоинства и недостатки структурного подхода.
10. Приведите пример структурной схемы ПО.
11. Для чего используют функциональные схемы?

12. Опишите основные элементы функциональных схем ПО.
13. В чем заключаются достоинства и недостатки использования функциональных схем?
14. Охарактеризуйте понятие UML.
15. Каковы преимущества использования UML?
16. Какие сущности описывают поведение системы?
17. Каковы типы сущностей в UML?
18. Перечислите виды диаграмм в UML.
19. Приведите примеры атрибутов ассоциаций. Что такое кратность ассоциаций?
20. Какие сущности обычно содержат диаграммы классов?
21. Постройте диаграмму деятельности для моделирования процесса проведения экзамена.
22. Какова цель разработки прототипов?
23. Перечислите виды прототипов.
24. Какие достоинства имеются у бумажных прототипов?
25. В каких случаях создается исследовательский прототип?
26. Для чего используются иллюстрированные сценарии прецедентов?
27. Перечислите общие принципы и правила создания удобного интерфейса.
28. Как используется цвет при разработке эргономичного интерфейса?
29. Перечислите основные требования к компоновке форм.
30. Как можно предотвратить появление ошибок при вводе?
31. Как правильно создать меню?

#### *Вопросы для самостоятельной проверки знаний*

1. Роль ПО и компьютеров в производстве, социальной жизни и науке.
2. Инженерия ПО
3. Проблемы разработки ПО
4. Технология разработки ПО
5. Системный подход к разработке ПО.
6. Временной и "пространственный" аспекты системного подхода
7. Этапы жизненного цикла ПО.
8. Каскадная модель жизненного цикла ПО.
9. Стандарты по разработке ПО.
10. Роль и назначение стандартов, требования стандартов
11. Три вида программных разработок с точки зрения технологии их создания
12. Параллельная разработка подсистем
13. Виды документов, выпускаемых на ПО по этапам разработки системы.
14. Задачи, решаемые на различных стадиях проектирования системы и ПО.
15. Стадии проектирования.
16. Цена ошибок проектирования.
17. Проектирование, основанное на моделировании (Model-Based Systems Engineering - MBSE)
18. CASE технологии разработки ПО
19. Задачи и результаты архитектурного проектирования ПО.
20. Технология Rational Rose, UML
21. Схема возможных вариантов совместного использования информации взаимодействующими процессами
22. Проектирование снизу-вверх и проектирование сверху-вниз.
23. Основные понятия структурного подхода к проектированию ПО.
24. Основные понятия объектно-ориентированного подхода к проектированию ПО.
25. Управление разработкой ПО.
26. Планирование разработки ПО.
27. Организация разработки ПО.

*Вопросы к зачету по дисциплине «Проектирование и архитектура программных систем»*

1. Технология проектирования программных систем (ПС).
2. CASE-технологии разработки ПС.
3. Структурный подход к проектированию ПС.
4. Инструментальные средства проектирования ПС.
5. Создание диаграмм потоков данных.
6. Объектно-ориентированный подход к проектированию ПС.
7. Унифицированный язык UML.
8. Варианты использования.
9. Диаграммы взаимодействия.
10. Диаграммы классов.
11. Диаграммы состояний объектов.
12. Диаграммы компонентов и размещения.
13. Что такое архитектура ПС, как ее определить?
14. Какие виды архитектур ПС характерны для современных программных систем.
15. Какие методы и средства моделирования архитектуры ПС вы знаете? Дайте их характеристику.
16. Двухуровневые и многоуровневые клиент-серверные архитектуры.
17. Архитектуры, основанные на потоках данных.
18. Архитектуры независимых компонентов.
19. CORBA – архитектура.
20. Сервис-ориентированные архитектуры.

**8. Учебно-методическое и информационное обеспечение дисциплины (модуля):**

**8.1 Основная литература**

1. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной. – М.: ИД «ФОРУМ»: ИНФРА-М., 2008.
2. Крылов Е.В., Острейковский В.А., Типикин Н.Г. Техника разработки программ: в 2 кн. Кн. 2. Технология, надежность и качество программного обеспечения. – М.: Высшая шк., 2008.
3. Назаров С.В. Архитектура и проектирование программных систем: Монография. – М.: ИНФРА-М, 2013.

**8.2 Дополнительная литература**

1. Shari Lawrence Pfleeger; Joanne M. Atlee. Software Engineering: Theory and Practice. Rand corporation Prentice Hall. 2005.
2. Зиглер К.. Методы проектирования программных систем: Пер. с англ. – М.: Мир, 1985.
3. Кулямин В. В. Технологии программирования. Компонентный подход. [Электронный ресурс]. Режим доступа: <http://lib.mdpu.org.ua/e-book/vstup/L/Jogolev.pdf>
4. Таненбаум Э. Современные операционные системы. 4-е изд. – СПб.: Питер, 2006.
5. Турский В. Методология программирования: пер. с англ. – М.: Мир, 1981.
6. Халл Э., Джексон К., Дик Д. Разработка и управление требованиями. Практическое руководство пользователя. Telelogic, 2005
7. Хьюз Дж., Мичтом Дж. Структурный подход к программированию: Пер. с англ. – М.: Мир, 1980.
8. Чернышов Л.Н. Среды разработки программного обеспечения: история и перспективы. [Электронный ресурс]. Режим доступа: <http://nit.miem.edu.ru/sbornik/2009-/plen/008.html>
9. Эберт К. Экскурс в историю программных технологий. – Открытые системы, №10, 2008.
10. Полис Г., Огастин Л., Мадхар Д. Разработка программных проектов: на основе Rational Unified Process (RUP). – М.: ООО «Бином-Пресс», 2009.

11. Ксензов М. Рефакторинг архитектуры программного обеспечения: выделение слоев. Труды ИПИРАН, 2006.

### 8.3 Программное обеспечение и Интернет-ресурсы

Программное обеспечение: для успешного освоения дисциплины, студент использует следующие программные средства:

- Программный продукт компании IBM Rational Software Architect
- Программный продукт компании IBM Rational RequisitePro.
- ОС Windows
- Пакет MS Office

Интернет-ресурсы:

- 1) <http://events.cnews.ru/events/>
- 2) <http://www.ibm.com/ru/events/>
- 3) <http://www.academy.it.ru/ru/about/actions/>
- 4) <http://www.techdays.ru/>
- 5) <https://msevents.microsoft.com/cui/default.aspx?culture=ru-ru>
- 6) <http://h41110.www4.hp.com/ru/institute.html>

### 8.4. Методические указания и материалы по видам занятий

- Инструкции к выполнению лабораторных работ по дисциплине
- Краткий курс лекций

### 9. Материально-техническое обеспечение дисциплины (модуля):

Учебный кабинет, лаборатория ИТО ИТИ.

### 10. Методические рекомендации по организации изучения дисциплины

Студенты, изучающие дисциплину проектирование и архитектуры программных систем, должны обратить внимание на широту и глубину знаний, требуемых для успешного освоения этой дисциплины. При этом на дисциплину отводится сравнительно небольшое количество часов аудиторных занятий. Отсюда следует, что результат успешного освоения дисциплины во многом определяется самостоятельной работой студентов. В связи с этим можно рекомендовать регулярный просмотр сайтов зарубежных и отечественных компаний, занимающихся разработкой программных систем.

Для наиболее успешного изучения дисциплины при подготовке к занятиям студентам следует самостоятельно изучить рекомендованную преподавателем литературу; ознакомиться с теоретическим материалом по теме курса, презентациями.

Многие ведущие программные компании (IBM, MS, HP и др.) реализуют программы академической инициативы, которые позволяют студентам получать бесплатно программные продукты, проходить курсы обучения, участвовать в семинарах и конференциях.

Самостоятельная работа студента является важной формой усвоения курса, поскольку в ходе ее выполнения слушатели не только овладевают знаниями по предмету, но и получают навыки самостоятельной работы с литературой и сетевыми информационными ресурсами. Для организации самостоятельной работы рекомендуется пользоваться учебными компьютерными классами и читальным залом библиотеки.

#### *Тест для самопроверки*

1. Приложение в технологии клиент/сервер предусматривает:

1. Однослойную архитектуру.
- +2. Трехслойную архитектуру.
3. Пятислойную архитектуру.

2. CASE - средства применяются прежде всего для:

- +1. Анализа и проектирования.
2. Тестирования.
3. Эксплуатации и сопровождения.



3. Общее хранилище данных в CASE - средствах называется:
  1. Банк данных.
  - +2. Репозиторий.
  3. Хранилище.
4. Классификация CASE - средств по типам:
  1. Отражает тип интерфейса.
  2. Определяет степень интегрированности по выполняемым функциям.
  - +3. Отражает функциональную ориентацию на те или иные процессы жизненного цикла.
5. Классификация CASE - средств по категориям:
  1. Отражает тип интерфейса.
  - +2. Определяет степень интегрированности по выполняемым функциям.
  3. Отражает функциональную ориентацию на те или иные процессы жизненного цикла.
6. Методология IDEF1X позволяет:
  1. Построить схему процессов, протекающих в системе.
  - +2. Построить модель данных.
  3. Описать алгоритм программы.
7. Объекты модели IDEF1X называются:
  - +1. Сущность.
  2. Атрибут.
  3. Экземпляр.
8. Наиболее длительный этап разработки программного обеспечения:
  1. Проектирование.
  2. Кодирование.
  - +3. Тестирование.
9. Стандартный язык описания разработки программных продуктов с использованием объектного подхода
  - +1. UML
  2. XML
  3. CASE-средства
  4. CMM
10. Многопользовательскую архитектуру реализуют системы, построенные по принципу
  1. Пакет программ
  2. Локальная сеть
  - +3. Клиент-сервер
  4. Программная система

Рабочая учебная программа по дисциплине «Проектирование и архитектура программных систем» составлена в соответствии с требованиями Федерального Государственного образовательного стандарта ВО по направлению 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ» и учебного плана по профилю подготовки «Разработка программно-информационных систем».

## ТЕХНОЛОГИЧЕСКАЯ КАРТА ДИСЦИПЛИНЫ

Курс 3

Семестр 6

Группа ИТ16ДР62ПИ1

Преподаватель – лектор **Белоконь О.С.**

Преподаватели, ведущие практические занятия – **Белоконь О.С.**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

| Наименование дисциплины/курса                                                                                     | Уровень образования<br>(бакалавриат, специалитет,<br>магистратура) | Статус дисциплины<br>в учебном<br>плане<br>(А, Б, В) | Количество<br>зачетных единиц       |                                      |
|-------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------|------------------------------------------------------|-------------------------------------|--------------------------------------|
| Проектирование и архитектура программных систем                                                                   | бакалавриат                                                        | А                                                    | 3                                   |                                      |
| СМЕЖНЫЕ ДИСЦИПЛИНЫ ПО УЧЕБНОМУ ПЛАНУ:                                                                             |                                                                    |                                                      |                                     |                                      |
| «Объектно-ориентированное программирование», «Введение в программную инженерию», «Разработка и анализ требований» |                                                                    |                                                      |                                     |                                      |
| БАЗОВЫЙ МОДУЛЬ (проверка знаний и умений по дисциплине)                                                           |                                                                    |                                                      |                                     |                                      |
| Тема,<br>задание или мероприятие<br>текущего контроля                                                             | Виды<br>текущей<br>аттестации                                      | Аудиторная<br>или внеаудиторная                      | Минимальное<br>количество<br>баллов | Максимальное<br>количество<br>баллов |
| Контрольная работа №1                                                                                             | КР1                                                                | аудиторная                                           | 10                                  | 20                                   |
| Лабораторная работа №1                                                                                            | ЛР1                                                                | аудиторная                                           | 5                                   | 10                                   |
| Лабораторная работа №2                                                                                            | ЛР2                                                                | аудиторная                                           | 5                                   | 10                                   |
| Лабораторная работа №3                                                                                            | ЛР3                                                                | аудиторная                                           | 5                                   | 10                                   |
| РУБЕЖНЫЙ КОНТРОЛЬ                                                                                                 | РК                                                                 |                                                      | 25                                  | 50                                   |
| Контрольная работа №2                                                                                             | КР2                                                                | аудиторная                                           | 10                                  | 20                                   |
| Лабораторная работа №4                                                                                            | ЛР4                                                                | аудиторная                                           | 5                                   | 10                                   |
| Лабораторная работа №5                                                                                            | ЛР5                                                                | аудиторная                                           | 5                                   | 10                                   |
| Лабораторная работа №6                                                                                            | ЛР6                                                                | аудиторная                                           | 5                                   | 10                                   |
| РУБЕЖНАЯ АТТЕСТАЦИЯ                                                                                               | РА                                                                 |                                                      | 25                                  | 50                                   |
| Итого                                                                                                             |                                                                    |                                                      | 50                                  | 100                                  |

Составитель, ст. преподаватель

О.С. Белоконь

Рабочая учебная программа рассмотрена методической комиссией инженерно-технического института протокол № 1 от «12» 09 2018 г. и признана соответствующей требованиям Федерального Государственного образовательного стандарта и учебного плана по направлению 09.03.04 «ПРОГРАММНАЯ ИНЖЕНЕРИЯ».

Председатель МК ИТИ

Е.И. Андрианова

Зав. кафедрой ПОВТ и АС, доцент

С.Г. Федорченко