

# МИНОБРНАУКИ РОССИИ

Воткинский филиал  
 Федерального государственного бюджетного образовательного  
 учреждения высшего образования  
 «Ижевский государственный технический университет имени М.Т. Калашникова»  
 (ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

УТВЕРЖДАЮ

Директор

И.А. Давыдов



2018 г.

## РАБОЧАЯ ПРОГРАММА

по дисциплине: Программная инженерия  
(наименование – полностью)  
 для направления: 09.03.01 «Информатика и вычислительная техника»  
(шифр, наименование – полностью)  
 по профилю «Автоматизированные системы обработки информации и управления»  
(наименование – полностью)  
 форма обучения: очная  
(очная, очно-заочная или заочная)

Общая трудоемкость дисциплины составляет: 6 зачетных единиц(ы)

Вид учебной работы	Всего часов	Семестры			
		3			
<b>Контактные занятия (всего)</b>	80	80			
В том числе:	-	-	-	-	-
Лекции	32	32			
Практические занятия (ПЗ)	16	16			
Семинары (С)					
Лабораторные работы (ЛР)	32	32			
<b>Самостоятельная работа (всего)</b>	136	136			
В том числе:	-	-	-	-	-
Курсовая работа	КР	КР			
Расчетно-графические работы					
Реферат					
<i>Другие виды самостоятельной работы</i>					
Вид промежуточной аттестации (зачет, экзамен)	Экзамен	Экзамен			
Общая трудоемкость	час	216	216		
	зач. ед.	6	6		

Кафедра «Организация вычислительных процессов и систем управления»

Составители Замятин Константин Игоревич, к.т.н., доцент,

Рабочая программа составлена на основании ФГОС ВО по направлению подготовки 09.03.01 «Информатика и вычислительная техника (уровень бакалавриата) №5 от 12.01.2016г. и утверждена на заседании кафедры

Протокол от « 19 » апреля 2018 г. № 04/18

Директор Воткинского филиала «ИжГТУ имени М.Т. Калашникова»

  
И.А. Давыдов  
« 19 » апреля 2018 г.

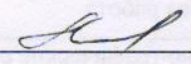
СОГЛАСОВАНО

Председатель учебно-методической комиссии  
по направлению 09.03.01 «Информатика и вычислительная техника»,  
профиль «Автоматизированные системы обработки  
информации и управления»

  
К.Б. Сентяков  
« 19 » апреля 2018 г.

Количество часов рабочей программы соответствует количеству часов рабочего учебного плана направления 09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления»

Ведущий специалист учебной части  
ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»

  
Соловьева Л.Н.  
« 19 » апреля 2018 г.

Аннотация к дисциплине

<b>Название дисциплины</b>		<b>Программная инженерия</b>					
<b>Номер</b>		<i>Академический год</i>		<i>семестр</i> 3			
<b>кафедра</b>		<i>Программа</i>	09.03.01 «Информатика и вычислительная техника», профиль «Автоматизированные системы обработки информации и управления»				
<b>Гарант модуля</b>		Замятин К.И., к.т.н., Касимов Д.Р., к.т.н.					
<b>Цели и задачи дисциплины, основные темы</b>		<p><b>Цели:</b> изучение теоретических основ современных технологий создания программных продуктов и получение практических навыков их реализации.</p> <p><b>Задачи:</b> формирование систематизированного представления о концепциях, моделях и принципах организации, положенных в основу «классических» технологий создания программных продуктов и современных семейств технологий; получение практической подготовки в области выбора и применения технологий создания программных продуктов для задач автоматизации обработки информации и управления.</p> <p><b>Знания:</b> система базовых понятий программной инженерии; классические и современные модели процесса разработки ПО; приемы реализации фаз жизненного цикла программного обеспечения.</p> <p><b>Умения:</b> выполнять основные виды работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества; ориентироваться в методах и средствах, используемых для создания ПО.</p> <p><b>Навыки:</b> сбора и анализа требований к ПО, планирования программного проекта, проектирования и реализации ПО, тестирования и документирования ПО; применения современных инструментальных средств поддержки процесса разработки ПО; командной разработки программных продуктов.</p> <p><b>Лекции (основные темы):</b> Введение в программную инженерию. Инженерия требований к ПО. Модели процесса разработки ПО. Проектирование ПО. Инструментальные средства поддержки процесса разработки ПО. Управление программными проектами. Методы обеспечения качества ПО.</p> <p><b>Лабораторные работы:</b> Оценка качества программного продукта. Разработка спецификации требований (SRS). Планирование программного проекта по методологии Scrum. Разработка UML-диаграммы состояний. Разработка UML-диаграммы деятельности. Разработка UML-диаграммы классов. Версионирование программного проекта с помощью системы контроля версий Git. Управление дефектами с помощью системы MantisBT.</p>					
<b>Основная литература</b>		<ol style="list-style-type: none"> <li>Носова, Л. С. Основы программной инженерии [Электронный ресурс] : учебно-методическое пособие / Л. С. Носова. — 2-е изд. — Электрон. текстовые данные. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 78 с. — 978-5-4486-0671-7. — Режим доступа: <a href="http://www.iprbookshop.ru/81488.html">http://www.iprbookshop.ru/81488.html</a></li> <li>Соловьев, Н. А. Введение в программную инженерию [Электронный ресурс] : учебное пособие / Н. А. Соловьев, Л. А. Юркевская. — Электрон. текстовые данные. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2017. — 112 с. — 978-5-7410-1685-5. — Режим доступа: <a href="http://www.iprbookshop.ru/71267.html">http://www.iprbookshop.ru/71267.html</a></li> <li>Кознов, Д. В. Введение в программную инженерию [Электронный ресурс] / Д. В. Кознов. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 306 с. — 2227-8397. — Режим доступа: <a href="http://www.iprbookshop.ru/52146.html">http://www.iprbookshop.ru/52146.html</a></li> <li>Липаев, В. В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс] : учебное пособие / В. В. Липаев. — Электрон. текстовые данные. — М. : МАКС Пресс, 2014. - 309 с. - 978-5-317-04750-4. — Режим доступа: <a href="http://www.iprbookshop.ru/27297.html">http://www.iprbookshop.ru/27297.html</a></li> </ol>					
<b>Технические средства</b>		Учебные аудитории для проведения занятий лекционного и семинарского типа, групповых и индивидуальных консультаций, текущего контроля успеваемости и промежуточной аттестации обучающихся, для самостоятельной работы студентов					
<b>Компетенции</b>		<b>Приобретаются студентами при освоении дисциплины</b>					
		<b>ОПК-2.</b> Способность осваивать методики использования программных средств для решения практических задач.					
		<b>ПК-1.</b> Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина".					
		<b>ПК-2.</b> Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.					
<b>Зачетных единиц</b>	6	<b>Форма проведения занятий</b>		<b>Лекции</b>	<b>Практические занятия</b>	<b>Лабораторные работы</b>	<b>Самостоятельная работа</b>
				32	16	32	136
<b>Виды контроля формы</b>	<b>Диф.зач /зач/ экз</b>	<b>КП/КР</b>	<b>Условие зачета дисциплины</b>	Получение оценок: 3,4,5 (Экз.) 3,4,5 (КР)	<b>Форма проведения самостоятельной работы</b>	Подготовка к практическим занятиям, лабораторным работам и экзамену, курсовая работа	
	Экз.	КР					
<b>Перечень модулей, знание которых необходимо для изучения модуля</b>			Информатика, Программирование				

## 1. Цели и задачи дисциплины:

**Целью** преподавания дисциплины является изучение теоретических основ современных технологий создания программных продуктов и получение практических навыков их реализации.

**Задачи** дисциплины:

- формирование систематизированного представления о концепциях, моделях и принципах организации, положенных в основу «классических» технологий создания программных продуктов и современных семейств технологий;
- получение практической подготовки в области выбора и применения технологий создания программных продуктов для задач автоматизации обработки информации и управления.

В результате изучения дисциплины студент должен

**знать:**

- систему базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план;
- классические и современные модели процесса разработки ПО;
- приемы реализации фаз жизненного цикла программного обеспечения;

**уметь:**

- выполнять основные виды работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества;
- ориентироваться в методах и средствах, используемых для создания ПО;

**владеть навыками:**

- сбора и анализа требований к ПО, планирования программного проекта, проектирования и реализации ПО, тестирования и документирования ПО;
- применения современных инструментальных средств поддержки процесса разработки ПО: систем контроля версий, систем управления дефектами, систем сборки и выпуска программных продуктов, систем непрерывной интеграции, средств автоматизации тестирования ПО;
- командной разработки программных продуктов.

## 2. Место дисциплины в структуре ООП:

Дисциплина относится к дисциплинам по выбору Блока 1 «Дисциплины (модули)» ООП.

Для изучения дисциплины студент должен

**знать:**

- языки программирования высокого уровня;
- технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах;
- основы объектно-ориентированного подхода к программированию;

**уметь:**

- разрабатывать и отлаживать программы не менее, чем на одном из алгоритмических языков высокого уровня;
- ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы;
- работать с современными системами программирования, включая объектно-ориентированные;

**владеть:**

- навыками использования пакетов прикладных программ;
- языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня;
- методами и средствами разработки и оформления технической документации.

Изучение дисциплины базируется на знаниях, полученных при изучении дисциплин: Информатика, Программирование.

### 3. Требования к результатам освоения дисциплины:

#### 3.1. Знания, приобретаемые в ходе изучения дисциплины

№ п/п З	Знания
1.	Система базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план
2.	Классические и современные модели процесса разработки ПО
3.	Приемы реализации фаз жизненного цикла программного обеспечения

#### 3.2. Умения, приобретаемые в ходе изучения дисциплины

№ п/п У	Умения
1.	Выполнение основных видов работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества
2.	Умение ориентироваться в методах и средствах, используемых для создания ПО

#### 3.3. Навыки, приобретаемые в ходе изучения дисциплины

№ п/п Н	Навыки
1.	Сбор и анализ требований к ПО, планирование программного проекта, проектирование и реализация ПО, тестирование и документирование ПО
2.	Применение современных инструментальных средств поддержки процесса разработки ПО: систем контроля версий, систем управления дефектами, систем сборки и выпуска программных продуктов, систем непрерывной интеграции, средств автоматизации тестирования ПО
3.	Командная разработка программных продуктов

#### 3.4. Компетенции, приобретаемые в ходе изучения дисциплины

Компетенции	Знания (№№ из 3.1)	Умения (№№ из 3.2)	Навыки (№№ из 3.3)
<b>ОПК-2.</b> Способность осваивать методики использования программных средств для решения практических задач.	1, 3	1, 2	1, 2
<b>ПК-1.</b> Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек - электронно-вычислительная машина".	1, 2, 3	1, 2	1, 2, 3
<b>ПК-2.</b> Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования.	1, 2, 3	1, 2	1, 2, 3

#### 4. Структура и содержание дисциплины (модуля)

##### 4.1. Разделы дисциплины и виды занятий

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды контактной работы, самостоятельная работа студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
				лек	прак	лаб	СРС*	
1	Введение в программную инженерию	3	1	2		2	4	Проверка остаточных знаний.
		3	2	2	2	2	4	Отчет по лабораторной работе № 1.
2	Инженерия требований к ПО	3	3	2		2	4	Определение темы курсовой работы.
		3	4	2	2	2	4	Техническое задание на курсовую работу.
		3	5	2		2	4	Отчет по лабораторной работе № 2.
3	Модели процесса разработки ПО	3	6	2	2	2	4	Глава «Сбор и анализ требований к ПО» пояснительной записки к курсовой работе.
		3	7	2		2	4	Контрольная работа № 1.
		3	8	2	2	2	4	Отчет по лабораторной работе № 3. 1-ая аттестация.
4	Проектирование ПО	3	9	2		2	4	Отчет по лабораторной работе № 4.
		3	10	2	2	2	4	Отчет по лабораторной работе № 5. Глава «Планирование программного проекта» пояснительной записки к курсовой работе.
5	Инструментальные средства поддержки процесса разработки ПО	3	11	2		2	4	Отчет по лабораторной работе № 6.
		3	12	2	2	2	4	Глава "Проектирование ПО" пояснительной записки к курсовой работе.
6	Управление программными проектами	3	13	2		2	4	2-ая аттестация.
		3	14	2	2	2	4	Отчет по лабораторной работе № 7.
7	Методы обеспечения качества ПО	3	15	2		2	4	Контрольная работа № 2.
		3	16	2	2	2	4	Отчет по лабораторной работе № 8.
	Курсовая работа						36	Пояснительная записка к курсовой работе. Защита курсовой работы. 3-ья аттестация.
	Экзамен						36	Экзамен.
	<b>ВСЕГО</b>			32	16	32	136	
	В том числе контроль самостоятельной работы				2			

\*включая курсовое проектирование

#### 4.2. Содержание разделов курса

№ п/п	Раздел дисциплины	Знания (номер из 3.1)	Умения (номер из 3.2)	Навыки (номер из 3.3)
1	<ol style="list-style-type: none"> <li>1. Основные понятия: программное обеспечение, программный продукт, жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО.</li> <li>2. Объекты изучения программной инженерии.</li> <li>3. Отличия программной индустрии от других индустрий.</li> <li>4. Характеристики качества ПО.</li> </ol>	1	1	1
2	<ol style="list-style-type: none"> <li>1. Выявление требований к ПО.</li> <li>2. Анализ требований.</li> <li>3. Документирование требований.</li> <li>4. Изменение требований.</li> <li>5. Управление требованиями.</li> </ol>	1, 3	1, 2	1, 3
3	<ol style="list-style-type: none"> <li>1. Классическая (водопадная) модель процесса разработки ПО.</li> <li>2. Прототипирование (макетирование).</li> <li>3. Инкрементная модель.</li> <li>4. Спиральная модель.</li> <li>5. Быстрая разработка приложений (RAD).</li> <li>6. Гибкие (agile) модели: <ul style="list-style-type: none"> <li>– Экстремальное программирование (XP).</li> <li>– Scrum.</li> </ul> </li> </ol>	2, 3	1, 2	1, 3
4	<ol style="list-style-type: none"> <li>1. Язык моделирования UML.</li> <li>2. Структурирование системы.</li> <li>3. Декомпозиция подсистем на модули.</li> <li>4. Модульность.</li> <li>5. Информационная закрытость.</li> <li>6. Связность модуля.</li> <li>7. Сцепление модулей.</li> </ol>	3	1, 2	1, 2, 3
5	<ol style="list-style-type: none"> <li>1. Системы контроля версий.</li> <li>2. Системы управления дефектами.</li> <li>3. Сборка и выпуск программного продукта.</li> <li>4. Непрерывная интеграция.</li> </ol>	1, 3	1, 2	1, 2, 3
6	<ol style="list-style-type: none"> <li>1. Ресурсы и роли в программных проектах.</li> <li>2. Проектный план.</li> </ol>	1, 3	1, 2	1, 2, 3
7	<ol style="list-style-type: none"> <li>1. Метод формальной дедуктивной верификации.</li> <li>2. Model Checking.</li> <li>3. Статический анализ.</li> <li>4. Тестирование.</li> <li>5. Аудит программного кода (code review).</li> </ol>	3	1, 2	1, 2, 3

#### 4.3. Наименование тем лабораторных работ, их содержание и объем в часах

№ п/п	№ раздела дисциплины	Наименование лабораторных работ	Трудоемкость (час)
1.	1	Оценка качества программного продукта	2
2.	2	Разработка спецификации требований (SRS)	6
3.	3	Планирование программного проекта по методологии Scrum	6
4.	4	Разработка UML-диаграммы состояний	2
5.	4	Разработка UML-диаграммы деятельности	2
6.	4	Разработка UML-диаграммы классов	2
7.	5	Версионирование программного проекта с помощью системы контроля версий Git	6
8.	5	Управление дефектами с помощью системы MantisBT	6
	<b>Всего</b>		<b>32</b>

#### 4.4. Наименование тем практических занятий, их содержание и объем в часах

№ п/п	№ раздела дисциплины	Наименование практических работ	Трудоемкость (час)
1.	1	Определение типа характеристик ПО	2
2.	2	Классификация требований к ПО	2
3.	3	Сбор требований к ПО методом прототипирования (макетирования)	2
4.	3	Работа над проектом по методологии Scrum	2
5.	4	Разработка UML-диаграмм	2
6.	5	Классификация дефектов ПО	2
7.	6	Составление плана-графика работ по проекту	2
8.	7	Оценка сложности программ	2
	<b>Всего</b>		<b>16</b>

### 5. Содержание самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

#### 5.1. Содержание самостоятельной работы

№ п/п	№ раздела дисциплины	Наименование тем	Трудоемкость (час)
1.	1	Статистические параметры проектов из различных областей программной индустрии	8
2.	2	Шаблоны спецификации требований (по ГОСТ, IEEE)	12
3.	3	Методологии разработки ПО Kanban, Feature Driven Development, Test Driven Development, Lean Software Development	12
4.	4	Паттерны объектно-ориентированного проектирования	8
5.	5	Расширенные возможности системы контроля версий Git	8
6.	6	Управление риском в программных проектах	8
7.	7	Библиотеки для автоматического тестирования ПО	8
8.		Выполнение курсовой работы	36
9.		Подготовка к экзамену	36
	<b>Всего</b>		<b>136</b>



**5.2.** Оценочные средства, используемые для текущего контроля успеваемости и промежуточной аттестации обучающихся по итогам освоения дисциплины, их виды и формы, требования к ним и шкалы оценивания приведены в приложении к рабочей программе дисциплины «Фонд оценочных средств по дисциплине Программная инженерия», которое оформляется в виде отдельного документа.

## **6. Учебно-методическое и информационное обеспечение дисциплины:**

### **а) основная литература**

Номер	Наименование книги	Год издания
1	Носова, Л. С. Основы программной инженерии [Электронный ресурс] : учебно-методическое пособие / Л. С. Носова. — 2-е изд. — Электрон. текстовые данные. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 78 с. — 978-5-4486-0671-7. — Режим доступа: <a href="http://www.iprbookshop.ru/81488.html">http://www.iprbookshop.ru/81488.html</a>	2019
2	Соловьев, Н. А. Введение в программную инженерию [Электронный ресурс] : учебное пособие / Н. А. Соловьев, Л. А. Юркевская. — Электрон. текстовые данные. — Оренбург : Оренбургский государственный университет, ЭБС АСВ, 2017. — 112 с. — 978-5-7410-1685-5. — Режим доступа: <a href="http://www.iprbookshop.ru/71267.html">http://www.iprbookshop.ru/71267.html</a>	2017
3	Кознов, Д. В. Введение в программную инженерию [Электронный ресурс] / Д. В. Кознов. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 306 с. — 2227-8397. — Режим доступа: <a href="http://www.iprbookshop.ru/52146.html">http://www.iprbookshop.ru/52146.html</a>	2016
4	Липаев, В. В. Программная инженерия сложных заказных программных продуктов [Электронный ресурс] : учебное пособие / В. В. Липаев. — Электрон. текстовые данные. — М. : МАКС Пресс, 2014. — 309 с. — 978-5-317-04750-4. — Режим доступа: <a href="http://www.iprbookshop.ru/27297.html">http://www.iprbookshop.ru/27297.html</a>	2014

### **б) дополнительная литература**

Номер	Наименование книги	Год издания
1	Носова, Л. С. Case-технологии и язык UML [Электронный ресурс] : учебно-методическое пособие / Л. С. Носова. — 2-е изд. — Электрон. текстовые данные. — Челябинск, Саратов : Южно-Уральский институт управления и экономики, Ай Пи Эр Медиа, 2019. — 67 с. — 978-5-4486-0670-0. — Режим доступа: <a href="http://www.iprbookshop.ru/81479.html">http://www.iprbookshop.ru/81479.html</a>	2019
2	Леоненков, А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose. Курс лекций [Электронный ресурс] : учебное пособие для студентов вузов, обучающихся по специальностям в области информационных технологий / А. В. Леоненков. — Электрон. текстовые данные. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Вузовское образование, 2017. — 318 с. — 978-5-4487-0081-1. — Режим доступа: <a href="http://www.iprbookshop.ru/67388.html">http://www.iprbookshop.ru/67388.html</a>	2017
3	Ружников, В. А. Экономика программной инженерии [Электронный ресурс] : учебное пособие / В. А. Ружников, М. А. Вержаковская, В. Ю. Аронов. — Электрон. текстовые данные. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2016. — 91 с. — 2227-8397. — Режим доступа:	2016

	<a href="http://www.iprbookshop.ru/73844.html">http://www.iprbookshop.ru/73844.html</a>	
4	Котляров, В. П. Основы тестирования программного обеспечения [Электронный ресурс] / В. П. Котляров. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 334 с. — 5-94774-406-4. — Режим доступа: <a href="http://www.iprbookshop.ru/62820.html">http://www.iprbookshop.ru/62820.html</a>	2016

#### **в) перечень ресурсов информационно-коммуникационной сети Интернет**

1. Электронно-библиотечная система IPRbooks  
<http://istu.ru/material/elektronno-bibliotechnaya-sistema-iprbooks>
2. Электронный каталог научной библиотеки ИжГТУ имени М.Т. Калашникова Web ИРБИС  
[http://94.181.117.43/cgi-bin/irbis64r\\_12/cgiirbis\\_64.exe?LNG=&C21COM=F&I21DBN=IBIS&P21DBN=IBIS](http://94.181.117.43/cgi-bin/irbis64r_12/cgiirbis_64.exe?LNG=&C21COM=F&I21DBN=IBIS&P21DBN=IBIS)

#### **г) программное обеспечение**

1. Система программирования Microsoft Visual Studio Community 2017
2. Система контроля версий Git (свободно распространяемое ПО).
3. Система управления дефектами MantisBT (свободно распространяемое ПО).
4. LibreOffice (свободно распространяемое ПО).

#### **д) методические указания**

1. Методические указания по дисциплине Программная инженерия / составители Е. А. Фролова. — М. : Московский технический университет связи и информатики, 2013. — 24 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/61752.html>.
2. Полетайкин, А. Н. Учебно-методическое пособие по выполнению лабораторных работ по дисциплине «Программная инженерия». Часть I. Реализация жизненного цикла программного обеспечения : учебно-методическое пособие / А. Н. Полетайкин. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2016. — 97 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/69565.html>
3. Миньков С.Л. Программная инженерия: курсовая работа. Томск, ТУСУР – 2016

#### **7. Материально-техническое обеспечение дисциплины:**

1. Специальные помещения – учебные аудитории для проведения занятий лекционного типа и практических занятий, оборудованные доской, экраном, проектором, столами, стульями.
2. Специальные помещения – учебные аудитории для проведения: занятий семинарского типа, групповых и индивидуальных консультаций, оборудованные доской, столами, стульями.
3. Специальные помещения – учебные аудитории для проведения лабораторных занятий, оборудованные доской, столами лабораторными, стульями, лабораторным оборудованием различной степени сложности.
4. Специальные помещения – учебные аудитории для выполнения курсовой работы, оборудованные доской, экраном, проектором, компьютерами с возможностью подключения к сети «Интернет», столами, стульями.
5. Специальные помещения – учебные аудитории для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся, оборудованные доской, экраном, проектором, компьютерами с возможностью подключения к сети «Интернет», столами, стульями.
6. Специальные помещения – учебные аудитории для организации и проведения самостоятельной работы студентов, оборудованные доской, компьютерами с возможностью подключения к сети «Интернет», столами, стульями.

**Лист утверждения рабочей программы дисциплины «Программная инженерия» на учебный  
год**

Рабочая программа дисциплины «Программная инженерия» утверждена на ведение учебного процесса в учебном году:

<b><i>Учебный год</i></b>	<b><i>«Согласовано»: заведующий кафедрой, ответственной за РПД (подпись и дата)</i></b>
2018- 2019	
2019- 2020	
2020- 2021	
2021 – 2022	
2022 - 2023	
2023 - 2024	
2024- 2025	

МИНОБРНАУКИ РОССИИ

Воткинский филиал  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Ижевский государственный технический университет  
имени М.Т. Калашникова»  
(ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

Кафедра Организация вычислительных процессов и систем управления  
(наименование кафедры)

УТВЕРЖДЕН

на заседании кафедры

«\_\_» \_\_\_\_\_ 2018 г., протокол № \_\_\_\_\_

Директор филиала

\_\_\_\_\_ Давыдов И.А.

(подпись)

# ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

«Программная инженерия»

(наименование дисциплины)

**09.03.01 «Информатика и вычислительная техника»**

(шифр и наименование направления)

«Автоматизированные системы обработки информации и управления»

(наименование профиля)

**Бакалавр**

\_\_\_\_\_ Квалификация (степень) выпускника

**Паспорт  
фонда оценочных средств  
по дисциплине «Программная инженерия»**

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контроли- руемой компетенции (или ее части)	Наименование оценочного сред- ства
1	Введение в программную ин- женерию	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
2	Инженерия требований к ПО	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
3	Модели процесса разработки ПО	ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
4	Проектирование ПО	ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
5	Инструментальные средства поддержки процесса разработ- ки ПО	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
6	Управление программными проектами	ПК-1, ПК-2	Работа на практических занятиях. Кон- трольная работа. Защита курсовой рабо- ты. Подготовка к экзамену.
7	Методы обеспечения качества ПО	ПК-1, ПК-2	Работа на практических занятиях. Кон- трольная работа. Защита курсовой рабо- ты. Подготовка к экзамену.

**1. Описания элементов ФОС**

**Наименование:** работа на практических занятиях: текущий контроль выполнения заданий.

**Представление в ФОС:** перечень заданий

**Варианты заданий:**

1. Определить тип характеристики ПО (функциональная пригодность, уровень производительности, совместимость, удобство использования, надежность, защищенность, сопровождаемость, переносимость). Примеры характеристик:

- программа «Калькулятор» выполняет операцию сложения не более чем за 1 мс;
- программа «Калькулятор» занимает не более 10 Мб оперативной памяти;
- в программе «Калькулятор» можно выполнять операцию сложения, вычитания, умножения, деления, возведения в степень, тригонометрические операции;
- в программе «Калькулятор» используется 6 знаков для представления дробной части числа;
- в программе «Калькулятор» можно обратиться к справке в формате СНМ;
- в программе «Калькулятор» можно скопировать результат в буфер обмена;
- программа «Калькулятор» работает под управлением операционной системы Windows;
- в программе «Калькулятор» предусмотрены следующие режимы работы: обычный, инженерный, программист, статистика;

- программа «Калькулятор» предоставляет СОМ-интерфейс для доступа внешних программ к ее функциям;
- программа «Калькулятор» обеспечивает ввод данных и команд как с помощью мыши, так и посредством клавиатуры;
- в программе «Калькулятор» кнопки цифр и операций имеют габаритный размер не менее 20 пикселей.

2. Классифицировать требование к ПО (функциональное требование, требование к качеству, требование к реализации, требование к аппаратному обеспечению, нет требования). Примеры требований:

- В госпитале пребывание пациентов может быть в обычной палате, отделении интенсивной терапии (ОИТ) или блоке контролируемой реабилитации (БКР).
- Пациент может быть в нескольких палатах во время нахождения в госпитале и в рамках одной палаты ему могут быть выделены разные кровати.
- Во время пребывания в палате может случиться так, что пациент время от времени находится не в кровати (например, когда выполняется операция). В этом случае функция мониторинга должна быть прервана. Она должна быть возобновлена при возвращении пациента в кровать.
- Для каждого пациента система отслеживает набор параметров. В зависимости от пациента, могут быть взяты на учет также некоторые параметры, специфичные конкретному пациенту.
- Новые параметры должны быть добавлены в функцию мониторинга при изменении медицинского оборудования.
- Система должна анализировать изображения, записанные видеокамерой для того, чтобы проверить эмоциональное состояние пациента.
- Все ОИТ- и БКР-пациенты должны подвергаться мониторингу. В случае БКР-пациентов обычно отслеживается меньше параметров, чем для ОИТ-пациентов. Во всех остальных аспектах работа с ними производится одинаково.
- Как правило, пациенты в обычных палатах не мониторятся.

3. Собрать требования к программе методом прототипирования (макетирования). Примеры программ:

- Семейный кошелек.
- Органайзер.
- Проигрыватель видеофайлов.
- Диктофон.
- Построитель графиков функций.

4. Разработать UML-диаграмму вариантов использования для описанной ниже программной системы:

- Во всем мире являются популярными соревнования по спортивному ориентированию. Лыжник должен за определенное время, имея у себя карту местности, пройти все контрольные точки, которые отмечены на этой карте. В настоящее время в этих контрольных точках стоят судьи, которые и отмечают факт прохождения спортсменом точек на карте. Однако в лесу холодно, и поэтому хочется заменить судей бездушными контрольными станциями, которые бы сообщали о факте прохождения контрольной точки на судейский пункт в момент, когда спортсмен прикасается к этой станции браслетом с чипом. Вся информация о ходе соревнования выводится на большое табло. Кроме того, за соревнованиями можно наблюдать через Интернет при условии создания аккаунта на сайте и подтверждения этого аккаунта администратором.

5. Определить категорию дефекта ПО (функциональный дефект, дефект требований, дефект документации и т.д.).

6. Построить диаграмму Гантта для проекта из курсовой работы.

7. Вычислить значения метрик Холстеда для заданной программы:

```
type
  TIntArray: array of Integer;
procedure BubbleSort(a: TIntArray);
var n, i, j, t: Integer;
```

```

begin
  n := Length(a);
  if n <= 1 then
    Exit;
  for i := 0 to n - 2 do
    for j := i + 1 to n - 1 do
      if a[i] > a[j]
      then begin
        t := a[i];
        a[i] := a[j];
        a[j] := t;
      end;
    end;
  end;
end;
procedure SelectionSort(var a: TIntArray);
var max, i, j, best_value, best_j: Integer;
begin
  max := Length(a) - 1;
  for i := 0 to max - 1 do
    begin
      best_value := a[i];
      best_j := i;
      for j := i + 1 to max do
        begin
          if a[j] > best_value
          then begin
            best_value := a[j];
            best_j := j;
          end;
        end;
      end;
      a[best_j] := a[i];
      a[i] := best_value;
    end;
  end;
end;

```

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** защита лабораторных работ

***Представление в ФОС:*** задания и требования к выполнению представлены в методических указаниях по дисциплине

***Варианты заданий:*** задания и требования к выполнению представлены в методических указаниях по дисциплине

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** контрольная работа

***Представление в ФОС:*** набор вариантов заданий

***Варианты заданий:***

1. Назовите гибкие (Agile) модели процесса разработки.
2. Когда следует использовать методологию разработки Waterfall вместо Scrum?
3. В каких моделях процесса разработки допускается изменение требований?
4. При использовании какого метода тестирования код программы доступен тестировщикам?
5. Начало какого этапа жизненного цикла ПО знаменует собой создание UML диаграммы классов?
6. Для чего применяются диаграммы вариантов использования (Use Case)?
7. Какую роль выполняет Scrum-мастер?
8. Каково назначение пятнадцатиминутных ежедневных совещаний (Daily Scrum Meeting) в Scrum?
9. Что такое фаза жизненного цикла ПО? Приведите основные фазы жизненного цикла ПО.
10. Каковы различия между Agile и традиционным управлением проектами (Waterfall)?

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** защита курсовых работ

**Представление в ФОС:** задания и требования к выполнению представлены в методических указаниях по дисциплине

**Варианты заданий:** задания и требования к выполнению представлены в методических указаниях по дисциплине

**Критерии оценки:**

Приведены в разделе 2

**Наименование:** экзамен

**Представление в ФОС:** перечень вопросов

**Перечень вопросов для проведения экзамена:**

1. Основные понятия дисциплины: программное обеспечение, программный продукт, жизненный цикл ПО, фаза жизненного цикла ПО, программная инженерия.
2. Модели процесса разработки ПО. Классическая (водопадная) модель разработки ПО.
3. Модели процесса разработки ПО. Прототипирование (макетирование).
4. Модели процесса разработки ПО. Инкрементная модель.
5. Модели процесса разработки ПО. Спиральная модель.
6. Модели процесса разработки ПО. Быстрая разработка приложений (RAD).
7. Модели процесса разработки ПО. Экстремальное программирование.
8. Модели процесса разработки ПО. SCRUM.
9. Инженерия требований. Требования в программных проектах. Проблемы определения требований. Виды требований. Свойства требований (требования к требованиям).
10. Разработка требований. Способы выявления требований. Проблемы выявления требований. Анализ требований (уточнение, структурирование, приоритезация).
11. Способы документирования требований. Шаблоны спецификации требований.
12. Причины изменения требований. Возможность изменения требований. Управление изменениями. Анализ влияния изменения требования.
13. Управление требованиями. Прослеживание требований.
14. Понятие «программный проект». Ресурсы в программных проектах. Виды ресурсов.
15. Роли участников в программных проектах.
16. Виды проектной деятельности в программных проектах.
17. Временные сущности программных проектов.
18. Визуализация проектного плана (диаграмма Гантта, диаграмма PERT). Наблюдение за программным проектом.
19. Понятие дефекта программного обеспечения. Характеристики дефектов.
20. Жизненный цикл дефекта.
21. Системы управления дефектами.
22. Предпосылки для версионирования ПО. Ветвление.
23. Системы контроля версий. Типы СКВ. Общие принципы организации.
24. Системы контроля версий. Типовые операции.
25. Сборка программных проектов. Проблемы при сборке программных проектов.
26. Сборка программных проектов. Окружение для сборки. Общие требования к системе сборки.
27. Непрерывная интеграция.
28. Качество программного обеспечения. Характеристики качества.
29. Оценка качества ПО. Программометрика. Метрики программного обеспечения (размера, Холстеда, Чепина, цикломатической сложности, MOOD).
30. Применение метрик ПО. Аудит программного кода.
31. Методы обеспечения качества ПО. Формальная верификация.
32. Методы обеспечения качества ПО. Метод проверки моделей.
33. Методы обеспечения качества ПО. Статический анализ ПО.
34. Методы обеспечения качества ПО. Тестирование ПО.

**Критерии оценки:**

Приведены в разделе 2



## Критерии оценки:

Компетенции	Дескрипторы	Вид, форма оценочного мероприятия	Уровень освоения компетенции			
			Компетенция освоена*			
			отлично	хорошо	удовлетворительно	неудовлетворительно
<p><b>ОПК-2.</b> Способность осваивать методики использования программных средств для решения практических задач.</p> <p><b>ПК-1.</b> Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек – электронно-вычислительная машина".</p> <p><b>ПК-2.</b> Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя</p>	<p>31: Система базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план.</p>	Контрольная работа	<p>Правильно выполнены все задания. Продемонстрирован высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий</p>	<p>Задания выполнены более чем наполовину. Присутствуют серьезные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Задания выполнены менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению</p>
	<p>32: Классические и современные модели процесса разработки ПО.</p> <p>33: Приемы реализации фаз жизненного цикла программного обеспечения.</p> <p>31: Система базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план.</p> <p>32: Классические и современные модели процесса разработки ПО.</p> <p>33: Приемы реализации фаз жизненного цикла программного обеспечения.</p>	экзамен	<p>заслуживает обучающийся, обнаруживший всестороннее, систематическое и глубокое знание учебного материала, предусмотренного программой, усвоивший основную литературу и знакомый с дополнительной литературой, рекомендованной программой.</p>	<p>заслуживает обучающийся, обнаруживший полное знание учебного материала, усвоивший основную литературу, рекомендованную в программе. Оценка "хорошо" выставляется обучающимся, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.</p>	<p>заслуживает обучающийся, обнаруживший знания основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, знакомых с основной литературой, рекомендованной программой. Оценка выставляется обучающимся, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.</p>	<p>выставляется обучающемуся, обнаружившему пробелы в знаниях основного учебного материала. Оценка ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании образовательного учреждения без дополнительных занятий по рассматриваемой дисциплине.</p>

<p>современные инструментальные средства и технологии программирования.</p>	<p>Н1: Сбор и анализ требований к ПО, планирование программного проекта, проектирование и реализация ПО, тестирование и документирование ПО</p> <p>Н2: Применение современных инструментальных средств поддержки процесса разработки ПО: систем контроля версий, систем управления дефектами, систем сборки и выпуска программных продуктов, систем непрерывной интеграции, средств автоматизации тестирования ПО</p> <p>Н3: Командная разработка программных продуктов</p> <p>У1: Выполнение основных видов работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества</p> <p>У2: Умение ориентироваться в методах и средствах, используемых для создания ПО</p>	<p>Защита лабораторных работ / курсовой работы</p>	<p>выставляется студенту, если задание выполнено в полном объеме с соблюдением необходимой последовательности. Студенты работают полностью самостоятельно: подбирают необходимые для выполнения предлагаемых работ в задании источники знаний, показывают необходимые для проведения практической работы теоретические знания, практические умения и навыки.</p>	<p>выставляется студенту, если задание выполнено в полном объеме и самостоятельно. Допускаются отклонения от необходимой последовательности выполнения, не влияющие на правильность конечного результата. Студенты используют указанные преподавателем источники знаний, включая страницы атласа, таблицы из приложения к учебнику, страницы из справочной литературы по предмету. Задание показывает знание учащегося основного теоретического материала и овладение умениями, необходимыми для самостоятельного выполнения работы. Могут быть неточности и небрежность в оформлении результатов работы.</p>	<p>выставляется студенту, если задание на работу выполняется и оформляется студентами при помощи преподавателя или хорошо подготовленных и уже выполненных на «отлично» данную работу студентов. На выполнение задания затрачивается много времени (можно дать возможность доделать работу дома). Студенты показывают знания теоретического материала, но испытывают затруднение при решении конкретной задачи.</p>	<p>выставляется, если студенты показывают плохое знание теоретического материала и отсутствие умения применить знания к решению практической задачи. Руководство и помощь со стороны преподавателя и хорошо подготовленных студентов неэффективны по причине плохой подготовки студента.</p>
	<p>Н1: Сбор и анализ требований к ПО, планирование программного проекта, проектирование и реализация ПО, тестирование и документирование ПО</p> <p>Н3: Командная разработка программных продуктов</p> <p>У1: Выполнение основных видов работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества</p>	<p>Работа на практических занятиях: текущий контроль выполнения заданий</p>	<p>Правильно выполнены все задания. Продемонстрирован высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий</p>	<p>Задания выполнены более чем наполовину. Присутствуют серьезные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Задания выполнены менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению</p>

МИНОБРНАУКИ РОССИИ

Воткинский филиал  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Ижевский государственный технический университет  
имени М.Т. Калашникова»  
(ВФ ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)

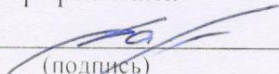
Кафедра Организация вычислительных процессов и систем управления  
(наименование кафедры)

УТВЕРЖДЕН

на заседании кафедры

«19» апр 2018 г., протокол № 04/18

Директор филиала

 Давыдов И.А.  
(подпись)

**ФОНД  
ОЦЕНОЧНЫХ СРЕДСТВ**

**ПО ДИСЦИПЛИНЕ**  
**«Программная инженерия»**  
(наименование дисциплины)

**09.03.01 «Информатика и вычислительная техника»**  
(шифр и наименование направления)

**«Автоматизированные системы обработки информации и управления»**  
(наименование профиля)

**Бакалавр**  
Квалификация (степень) выпускника

**Паспорт  
фонда оценочных средств  
по дисциплине «Программная инженерия»**

№ П/П	Контролируемые разделы (темы) дисциплины*	Код контроли- руемой компетенции (или ее части)	Наименование оценочного сред- ства
1	Введение в программную ин- женерию	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
2	Инженерия требований к ПО	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
3	Модели процесса разработки ПО	ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
4	Проектирование ПО	ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
5	Инструментальные средства поддержки процесса разработ- ки ПО	ОПК-2, ПК-1, ПК-2	Работа на практических занятиях. Защи- та лабораторных работ. Контрольная ра- бота. Защита курсовой работы. Подго- товка к экзамену.
6	Управление программными проектами	ПК-1, ПК-2	Работа на практических занятиях. Кон- трольная работа. Защита курсовой рабо- ты. Подготовка к экзамену.
7	Методы обеспечения качества ПО	ПК-1, ПК-2	Работа на практических занятиях. Кон- трольная работа. Защита курсовой рабо- ты. Подготовка к экзамену.

**1. Описания элементов ФОС**

**Наименование:** работа на практических занятиях: текущий контроль выполнения за-  
даний.

**Представление в ФОС:** перечень заданий

**Варианты заданий:**

1. Определить тип характеристики ПО (функциональная пригодность, уровень про-  
изводительности, совместимость, удобство использования, надежность, защищен-  
ность, сопровождаемость, переносимость). Примеры характеристик:

- программа «Калькулятор» выполняет операцию сложения не более чем за 1 мс;
- программа «Калькулятор» занимает не более 10 Мб оперативной памяти;
- в программе «Калькулятор» можно выполнять операцию сложения, вычитания, умножения, деления, возведения в степень, тригонометрические операции;
- в программе «Калькулятор» используется 6 знаков для представления дробной части числа;
- в программе «Калькулятор» можно обратиться к справке в формате СНМ;
- в программе «Калькулятор» можно скопировать результат в буфер обмена;
- программа «Калькулятор» работает под управлением операционной системы Windows;
- в программе «Калькулятор» предусмотрены следующие режимы работы: обычный, инже-  
нерный, программист, статистика;

- программа «Калькулятор» предоставляет СОМ-интерфейс для доступа внешних программ к ее функциям;
- программа «Калькулятор» обеспечивает ввод данных и команд как с помощью мыши, так и посредством клавиатуры;
- в программе «Калькулятор» кнопки цифр и операций имеют габаритный размер не менее 20 пикселей.

2. Классифицировать требование к ПО (функциональное требование, требование к качеству, требование к реализации, требование к аппаратному обеспечению, нет требования). Примеры требований:

- В госпитале пребывание пациентов может быть в обычной палате, отделении интенсивной терапии (ОИТ) или блоке контролируемой реабилитации (БКР).
- Пациент может быть в нескольких палатах во время нахождения в госпитале и в рамках одной палаты ему могут быть выделены разные кровати.
- Во время пребывания в палате может случиться так, что пациент время от времени находится не в кровати (например, когда выполняется операция). В этом случае функция мониторинга должна быть прервана. Она должна быть возобновлена при возвращении пациента в кровать.
- Для каждого пациента система отслеживает набор параметров. В зависимости от пациента, могут быть взяты на учет также некоторые параметры, специфичные конкретному пациенту.
- Новые параметры должны быть добавлены в функцию мониторинга при изменении медицинского оборудования.
- Система должна анализировать изображения, записанные видеокамерой для того, чтобы проверить эмоциональное состояние пациента.
- Все ОИТ- и БКР-пациенты должны подвергаться мониторингу. В случае БКР-пациентов обычно отслеживается меньше параметров, чем для ОИТ-пациентов. Во всех остальных аспектах работа с ними производится одинаково.
- Как правило, пациенты в обычных палатах не мониторятся.

3. Собрать требования к программе методом прототипирования (макетирования). Примеры программ:

- Семейный кошелек.
- Органайзер.
- Проигрыватель видеофайлов.
- Диктофон.
- Построитель графиков функций.

4. Разработать UML-диаграмму вариантов использования для описанной ниже программной системы:

- Во всем мире являются популярными соревнования по спортивному ориентированию. Лыжник должен за определенное время, имея у себя карту местности, пройти все контрольные точки, которые отмечены на этой карте. В настоящее время в этих контрольных точках стоят судьи, которые и отмечают факт прохождения спортсменом точек на карте. Однако в лесу холодно, и поэтому хочется заменить судей бездушными контрольными станциями, которые бы сообщали о факте прохождения контрольной точки на судейский пункт в момент, когда спортсмен прикасается к этой станции браслетом с чипом. Вся информация о ходе соревнования выводится на большое табло. Кроме того, за соревнованиями можно наблюдать через Интернет при условии создания аккаунта на сайте и подтверждения этого аккаунта администратором.

5. Определить категорию дефекта ПО (функциональный дефект, дефект требований, дефект документации и т.д.).

6. Построить диаграмму Гантта для проекта из курсовой работы.

7. Вычислить значения метрик Холстеда для заданной программы:

```
type
  TIntArray: array of Integer;
procedure BubbleSort(a: TIntArray);
var n, i, j, t: Integer;
```

```

begin
  n := Length(a);
  if n <= 1 then
    Exit;
  for i := 0 to n - 2 do
    for j := i + 1 to n - 1 do
      if a[i] > a[j]
      then begin
        t := a[i];
        a[i] := a[j];
        a[j] := t;
      end;
    end;
  end;
end;
procedure SelectionSort(var a: TIntArray);
var max, i, j, best_value, best_j: Integer;
begin
  max := Length(a) - 1;
  for i := 0 to max - 1 do
    begin
      best_value := a[i];
      best_j := i;
      for j := i + 1 to max do
        begin
          if a[j] > best_value
          then begin
            best_value := a[j];
            best_j := j;
          end;
        end;
      end;
      a[best_j] := a[i];
      a[i] := best_value;
    end;
  end;
end;

```

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** защита лабораторных работ

***Представление в ФОС:*** задания и требования к выполнению представлены в методических указаниях по дисциплине

***Варианты заданий:*** задания и требования к выполнению представлены в методических указаниях по дисциплине

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** контрольная работа

***Представление в ФОС:*** набор вариантов заданий

***Варианты заданий:***

1. Назовите гибкие (Agile) модели процесса разработки.
2. Когда следует использовать методологию разработки Waterfall вместо Scrum?
3. В каких моделях процесса разработки допускается изменение требований?
4. При использовании какого метода тестирования код программы доступен тестировщикам?
5. Начало какого этапа жизненного цикла ПО знаменует собой создание UML диаграммы классов?
6. Для чего применяются диаграммы вариантов использования (Use Case)?
7. Какую роль выполняет Scrum-мастер?
8. Каково назначение пятнадцатиминутных ежедневных совещаний (Daily Scrum Meeting) в Scrum?
9. Что такое фаза жизненного цикла ПО? Приведите основные фазы жизненного цикла ПО.
10. Каковы различия между Agile и традиционным управлением проектами (Waterfall)?

***Критерии оценки:***

Приведены в разделе 2

***Наименование:*** защита курсовых работ

**Представление в ФОС:** задания и требования к выполнению представлены в методических указаниях по дисциплине

**Варианты заданий:** задания и требования к выполнению представлены в методических указаниях по дисциплине

**Критерии оценки:**

Приведены в разделе 2

**Наименование:** экзамен

**Представление в ФОС:** перечень вопросов

**Перечень вопросов для проведения экзамена:**

1. Основные понятия дисциплины: программное обеспечение, программный продукт, жизненный цикл ПО, фаза жизненного цикла ПО, программная инженерия.
2. Модели процесса разработки ПО. Классическая (водопадная) модель разработки ПО.
3. Модели процесса разработки ПО. Прототипирование (макетирование).
4. Модели процесса разработки ПО. Инкрементная модель.
5. Модели процесса разработки ПО. Спиральная модель.
6. Модели процесса разработки ПО. Быстрая разработка приложений (RAD).
7. Модели процесса разработки ПО. Экстремальное программирование.
8. Модели процесса разработки ПО. SCRUM.
9. Инженерия требований. Требования в программных проектах. Проблемы определения требований. Виды требований. Свойства требований (требования к требованиям).
10. Разработка требований. Способы выявления требований. Проблемы выявления требований. Анализ требований (уточнение, структурирование, приоритезация).
11. Способы документирования требований. Шаблоны спецификации требований.
12. Причины изменения требований. Возможность изменения требований. Управление изменениями. Анализ влияния изменения требования.
13. Управление требованиями. Прослеживание требований.
14. Понятие «программный проект». Ресурсы в программных проектах. Виды ресурсов.
15. Роли участников в программных проектах.
16. Виды проектной деятельности в программных проектах.
17. Временные сущности программных проектов.
18. Визуализация проектного плана (диаграмма Гантта, диаграмма PERT). Наблюдение за программным проектом.
19. Понятие дефекта программного обеспечения. Характеристики дефектов.
20. Жизненный цикл дефекта.
21. Системы управления дефектами.
22. Предпосылки для версионирования ПО. Ветвление.
23. Системы контроля версий. Типы СКВ. Общие принципы организации.
24. Системы контроля версий. Типовые операции.
25. Сборка программных проектов. Проблемы при сборке программных проектов.
26. Сборка программных проектов. Окружение для сборки. Общие требования к системе сборки.
27. Непрерывная интеграция.
28. Качество программного обеспечения. Характеристики качества.
29. Оценка качества ПО. Программометрика. Метрики программного обеспечения (размера, Холстеда, Чепина, цикломатической сложности, MOOD).
30. Применение метрик ПО. Аудит программного кода.
31. Методы обеспечения качества ПО. Формальная верификация.
32. Методы обеспечения качества ПО. Метод проверки моделей.
33. Методы обеспечения качества ПО. Статический анализ ПО.
34. Методы обеспечения качества ПО. Тестирование ПО.

**Критерии оценки:**

Приведены в разделе 2

## Критерии оценки:

Компетенции	Дескрипторы	Вид, форма оценочного мероприятия	Уровень освоения компетенции			
			Компетенция освоена*			
			отлично	хорошо	удовлетворительно	неудовлетворительно
<p><b>ОПК-2.</b> Способность осваивать методики использования программных средств для решения практических задач.</p> <p><b>ПК-1.</b> Способность разрабатывать модели компонентов информационных систем, включая модели баз данных и модели интерфейсов "человек – электронно-вычислительная машина".</p> <p><b>ПК-2.</b> Способность разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя</p>	<p>31: Система базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план.</p> <p>32: Классические и современные модели процесса разработки ПО.</p> <p>33: Приемы реализации фаз жизненного цикла программного обеспечения.</p>	Контрольная работа	<p>Правильно выполнены все задания. Продемонстрирован высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий</p>	<p>Задания выполнены более чем наполовину. Присутствуют серьезные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Задания выполнены менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению</p>
	<p>31: Система базовых понятий программной инженерии: программное обеспечение (ПО), жизненный цикл ПО, фаза жизненного цикла ПО, программный проект, качество ПО, модель процесса разработки ПО, требование к ПО, ресурсы и роли в программных проектах, проектный план.</p> <p>32: Классические и современные модели процесса разработки ПО.</p> <p>33: Приемы реализации фаз жизненного цикла программного обеспечения.</p>	экзамен	<p>заслуживает обучающийся, обнаруживший всестороннее, систематическое и глубокое знание учебного материала, предусмотренного программой, усвоивший основную литературу и знакомый с дополнительной литературой, рекомендованной программой.</p>	<p>заслуживает обучающийся, обнаруживший полное знание учебного материала, усвоивший основную литературу, рекомендованную в программе. Оценка "хорошо" выставляется обучающимся, показавшим систематический характер знаний по дисциплине и способным к их самостоятельному пополнению и обновлению в ходе дальнейшей учебной работы и профессиональной деятельности.</p>	<p>заслуживает обучающийся, обнаруживший знания основного учебного материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профессии, знакомых с основной литературой, рекомендованной программой. Оценка выставляется обучающимся, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.</p>	<p>выставляется обучающемуся, обнаружившему пробелы в знаниях основного учебного материала. Оценка ставится обучающимся, которые не могут продолжить обучение или приступить к профессиональной деятельности по окончании образовательного учреждения без дополнительных занятий по рассматриваемой дисциплине.</p>



<p>современные инструментальные средства и технологии программирования.</p>	<p>Н1: Сбор и анализ требований к ПО, планирование программного проекта, проектирование и реализация ПО, тестирование и документирование ПО</p> <p>Н2: Применение современных инструментальных средств поддержки процесса разработки ПО: систем контроля версий, систем управления дефектами, систем сборки и выпуска программных продуктов, систем непрерывной интеграции, средств автоматизации тестирования ПО</p> <p>Н3: Командная разработка программных продуктов</p> <p>У1: Выполнение основных видов работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества</p> <p>У2: Умение ориентироваться в методах и средствах, используемых для создания ПО</p>	<p>Защита лабораторных работ / курсовой работы</p>	<p>выставляется студенту, если задание выполнено в полном объеме с соблюдением необходимой последовательности. Студенты работают полностью самостоятельно: подбирают необходимые для выполнения предлагаемых работ в задании источники знаний, показывают необходимые для проведения практической работы теоретические знания, практические умения и навыки.</p>	<p>выставляется студенту, если задание выполнено в полном объеме и самостоятельно. Допускаются отклонения от необходимой последовательности выполнения, не влияющие на правильность конечного результата. Студенты используют указанные преподавателем источники знаний, включая страницы атласа, таблицы из приложения к учебнику, страницы из справочной литературы по предмету. Задание показывает знание учащегося основного теоретического материала и овладение умениями, необходимыми для самостоятельного выполнения работы. Могут быть неточности и небрежность в оформлении результатов работы.</p>	<p>выставляется студенту, если задание на работу выполняется и оформляется студентами при помощи преподавателя или хорошо подготовленных и уже выполненных на «отлично» данную работу студентов. На выполнение задания затрачивается много времени (можно дать возможность доделать работу дома). Студенты показывают знания теоретического материала, но испытывают затруднение при решении конкретной задачи.</p>	<p>выставляется, если студенты показывают плохое знание теоретического материала и отсутствие умения применить знания к решению практической задачи. Руководство и помощь со стороны преподавателя и хорошо подготовленных студентов неэффективны по причине плохой подготовки студента.</p>
	<p>Н1: Сбор и анализ требований к ПО, планирование программного проекта, проектирование и реализация ПО, тестирование и документирование ПО</p> <p>Н3: Командная разработка программных продуктов</p> <p>У1: Выполнение основных видов работ в рамках жизненного цикла ПО: формирование и анализ требований, проектирование и реализация ПО, документирование ПО, анализ свойств ПО и контроль их качества</p>	<p>Работа на практических занятиях: текущий контроль выполнения заданий</p>	<p>Правильно выполнены все задания. Продемонстрирован высокий уровень владения материалом. Проявлены превосходные способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован хороший уровень владения материалом. Проявлены средние способности применять знания и умения к выполнению конкретных заданий</p>	<p>Задания выполнены более чем наполовину. Присутствуют серьезные ошибки. Продемонстрирован удовлетворительный уровень владения материалом. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий.</p>	<p>Задания выполнены менее чем наполовину. Продемонстрирован неудовлетворительный уровень владения материалом. Проявлены недостаточные способности применять знания и умения к выполнению</p>