



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

УТВЕРЖДАЮ

И.о. проректора

А.В. Троицкий

«__» _____ 2023г.

***ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН***

**РАБОЧАЯ ПРОГРАММА
ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ И СИСТЕМЫ КОЛЛЕКТИВНОЙ РАЗРАБОТКИ
ПРОГРАММ»**

Направление подготовки: 01.03.02. Прикладная математика и информатика

Профиль: Программирование. Математическое моделирование

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023


Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: Вилисов В.Я. Рабочая программа дисциплины: Технологии и системы коллективной разработки программ. – Королев МО: «Технологический Университет», 2023г.

Рецензент: д.э.н. проф. Мищенко А.В.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки бакалавров 01.03.02 «Прикладная математика и информатика» и Учебного плана, утвержденного Ученым советом Университета. Протокол № 9 от 11.04.2023 г.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Бугай И.В. к.т.н., доцент 			
Год утверждения (переподтверждения)	2023			
Номер и дата протокола заседания кафедры	№8 от 15.03.2023			

Рабочая программа согласована:

Руководитель ОПОП ВО _____  И.В. Бугай, к.т.н., доцент

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переподтверждения)	2023			
Номер и дата протокола заседания УМС	№5 от 11.04.2023			

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП ВО

Целью изучения дисциплины является:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

профессиональные компетенции (ПК):

- Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники (ПК-5);
- Способность принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их коллективной разработки ПО (ПК-7).

Основными **задачами** дисциплины являются:

1. Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
2. Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

Показатель освоения компетенции отражают следующие индикаторы:

Необходимые знания:

- Знать основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития
- Знать принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения
- Знать методы и средства проектирования программного обеспечения, баз данных, программных интерфейсов

Необходимые умения:

- Уметь использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта
- Уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения;

Трудовые действия:

- Иметь практический опыт применения указанных выше методов и технологий
- Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Технологии и системы коллективной разработки программ» относится к дисциплинам части, формируемой участниками образовательных отношений, основной профессиональной образовательной программы подготовки бакалавров по направлению подготовки 01.03.02. «Прикладная математика и информатика».

Изучение данной дисциплины базируется на знаниях, полученных в процессе изучения дисциплин: «Технологии и среды программирования», «Моделирование информационных процессов и систем», «Языки высокого уровня», «Геометрическое моделирование и компьютерная графика» и компетенциях: ПК-2, ПК-4, ПК-5, ОПК-2, ОПК-5, ПК-3.

Знания и компетенции, полученные при освоении дисциплины, являются базовыми при выполнении выпускной квалификационной работы бакалавра.

3. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины для студентов очной формы составляет 2 зачетных единицы, 72 часа. Практическая подготовка обучающихся составляет 24 часа.

Таблица 1

Виды занятий	Всего часов	Семестр седьмой	Семестр
ОЧНАЯ ФОРМА ОБУЧЕНИЯ			
Общая трудоемкость	72	72	
Аудиторные занятия	48	48	
Лекции (Л)	16	16	
Практические занятия (ПЗ)	32	32	
Лабораторные работы (ЛР)	-	-	
Практическая подготовка	24	24	
Самостоятельная работа	24	24	
Курсовые, расчетно-графические работы	-	-	
Расчетно-графические работы	-	-	
Контрольная работа	+	+	
Текущий контроль знаний	Тест	Тест	
Вид итогового контроля	зачет	зачет	
ЗАОЧНАЯ ФОРМА НЕ ПРЕДУСМОТРЕНА УЧЕБНЫМ ПЛАНОМ			

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Лабораторные занятия, час	Занятия в интерактивной форме, час	Практическая подготовка	Код компетенций
Тема 1. Методология коллективной разработки ПО	4	8	-	2	8	ПК-5,7
Тема 2. Планирование, мониторинг и управление работами коллектива разработчиков ПО	6	12	-	8	8	ПК-5,7
Тема 3. Мониторинг версий и непрерывная интеграция проекта ПО	6	12	-	6	8	ПК-5,7
Итого:	16	32	-	16	24	

4.2. Содержание тем дисциплины

Тема 1. Методология коллективной разработки ПО.

1.1. Каскадная модель. Этапы: определение требований, проектирование, конструирование (реализация, кодирование), воплощение, тестирование и отладка (верификация), инсталляция, поддержка.

1.2. Гибкая методология разработки. Принципы: ранняя и бесперебойная поставка ПО; возможность изменения требований даже в конце разработки; частая поставка версий ПО; ежедневное общение заказчика с разработчиками; проектом занимаются мотивированные личности; контакты лицом к лицу; постоянное улучшение мастерства дизайнера; не делать лишней работы; самоорганизующаяся команда; постоянная адаптация к изменяющимся обстоятельствам.

1.3. Обзор инструментов поддержки методологии коллективной работы (планирования, мониторинга, управления работами коллектива, управление версиями и мониторинг ПО, непрерывная интеграция проекта, средства виртуализации окружения при работе в разных средах).

Тема 2. Планирование, мониторинг и управление работами коллектива разработчиков ПО.

2.1. Технология Atlassian - JIRA Software (планирование, отслеживание, выпуск ПО). Планирование: создание пользовательских историй и задач; планирование спринтов, распределение заданий в команде разработчиков. Отслеживание: расстановка приоритетов; обсуждение работы команды; обеспечение абсолютной прозрачности. Выпуск ПО: выпуск новых версий; обеспечение актуальности

информации о всех элементах проекта ПО. Отчеты: визуализация данных о текущем состоянии проекта в реальном времени.

2.2. Средство хранения проектной документации от Atlassian - Confluence: единая рабочая среда, систематизация, комментарии, поддержка совещаний, интеграция с JIRA.

Тема 3. Мониторинг версий и непрерывная интеграция проекта ПО.

3.1. Мониторинг версий ПО в системах SubVersion и Git.

3.2. Непрерывная интеграция проекта (Jenkins). Частые автоматизированных сборок проекта для выявления и решения интеграционных проблем, раннего обнаружения и устранения ошибок и противоречий в проекте ПО.

Автоматизированная интеграция проекта: получение исходного кода из репозитория; сборка проекта; выполнение тестов; развертывание готового проекта; отправка отчетов. Разновидности сборки проекта: по внешнему запросу, по расписанию, по факту обновления репозитория и по другим критериям.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

Методические указания для обучающихся по освоению дисциплины.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Технологии и системы коллективной разработки программ» приведен в Приложении 1 к настоящей рабочей программе.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Соснин П. И. Человеко-компьютерное взаимодействие: учебное пособие / П. И. Соснин, В. В. Валюх. — Ульяновск: УлГТУ, 2020. — 119 с. — ISBN 978-5-9795-2020-9. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/259775>
2. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.: КУРС, НИЦ ИНФРА-М, 2016. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6, URL: <http://znanium.com/catalog/product/551224>.

Дополнительная литература:

1. Интеллектуальные системы проектирования и управления техническими объектами: учебное электронное издание: учебное пособие: в 4 частях / В. А. Немтинов, С. В. Карпушкин, В. Г. Мокрозуб [и др.]; Тамбовский государственный технический университет. – Тамбов: Тамбовский государственный технический университет (ТГТУ), 2018. – Часть 3. – 153 с.: табл., граф., схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=570332>

2. Арыков, С.Б. Параллельное программирование над общей памятью: OpenMP: [16+] / С.Б. Арыков, М.А. Городничев, Г.А. Щукин; Новосибирский государственный технический университет. – Новосибирск: Новосибирский государственный технический университет, 2019. – 95 с.: ил., табл. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=576119>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

<http://www.znaniyum.com/> - электронно-библиотечная система

<http://www.e.lanbook.com/> - ЭБС Издательства "ЛАНЬ"

<http://www.rucont.ru/> - электронно-библиотечная система

<http://www.biblioclub.ru/> - университетская библиотека онлайн.

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2 к настоящей рабочей программе.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения: MSOffice

Информационные справочные системы: Электронные ресурсы образовательной среды Университета

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам и к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

Практические занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам, к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ
«ТЕХНОЛОГИИ И СИСТЕМЫ КОЛЛЕКТИВНОЙ РАЗРАБОТКИ
ПРОГРАММ»**

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Программирование, математическое моделирование

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся приобретает:		
				Необходимые знания	Необходимые умения	Трудовые действия
1.	ПК-5	Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники	Тема 1-3.	Знать основные методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программных продуктов и программных комплексов, их сопровождения, администрирования и развития	Уметь использовать методы проектирования и производства программного продукта, принципы построения, структуры и приемы работы с инструментальными средствами, поддерживающими создание программного продукта	Иметь практический опыт применения указанных выше методов и технологий
2.	ПК-7	Способность принимать участие в управлении проектами создания информационных систем и программных комплексов на стадиях их коллективной разработки ПО	Тема 1-3.	Знать принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения Знать методы и средства проектирования программного обеспечения, баз данных, программных интерфейсов	Уметь использовать существующие типовые решения и шаблоны проектирования программного обеспечения	Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Этапы и показатель оценивания компетенции	Шкала и критерии оценки
ПК-5, ПК-7	Тест	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 90% правильных ответов</p> <p>Б) частично сформирована: •компетенция освоена на продвинутом уровне – 70% правильных ответов; •компетенция освоена на базовом уровне – от 51% правильных ответов;</p> <p>В) не сформирована (компетенция не освоена) – менее 50% правильных ответов</p>	<p>Проводится письменно</p> <p>Время, отведенное на процедуру – 30 мин.</p> <p>Неявка 0 баллов.</p> <p>Критерии оценки определяются процентным соотношением.</p> <p>Неудовлетворительно – менее 50% правильных ответов.</p> <p>Удовлетворительно – от 51% правильных ответов.</p> <p>Хорошо – от 70%.</p> <p>Отлично – от 90%.</p> <p>Максимальная оценка – 5 баллов.</p>
	Выполнение контрольной работы	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована: •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла;</p> <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>При определении сформированности компетенций критериями оценивания выступают методические рекомендации, разработанные по дисциплине для данного вида.</p>

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1. Типовые вопросы, выносимые на тестирование

1. Что включается в UML?

- (?) Диаграмма структуры данных.
- (!) Диаграмма классов.
- (?) Функциональная диаграмма.

2. Для чего предназначена диаграмма вариантов использования?

(?) Моделирования статической структуры классов системы и связей между ними.

(!) Моделирования бизнес-процессов организации и требований к создаваемой системе.

(?) Моделирования поведения объектов системы при переходе из одного состояния в другое.

(?) Моделирования физической архитектуры системы.

(?) Ни один из представленных вариантов не является верным.

3. Для чего предназначена диаграмма классов?

(!) Моделирования статической структуры классов системы и связей между ними.

(?) Моделирования бизнес-процессов организации и требований к создаваемой системе.

(?) Моделирования поведения объектов системы при переходе из одного состояния в другое.

(?) Моделирования физической архитектуры системы.

Ни один из представленных вариантов не является верным.

4. Для чего предназначена диаграмма состояний?

(?) Моделирования статической структуры классов системы и связей между ними.

(?) Моделирования бизнес-процессов организации и требований к создаваемой системе.

(!) Моделирования поведения объектов системы при переходе из одного состояния в другое.

(?) Моделирования физической архитектуры системы.

Ни один из представленных вариантов не является верным.

5. Для чего предназначена диаграмма развертывания?

(?) Моделирования статической структуры классов системы и связей между ними.

(?) Моделирования бизнес-процессов организации и требований к создаваемой системе.

(?) Моделирования поведения объектов системы при переходе из одного состояния в другое.

(!) Моделирования физической архитектуры системы.

(?) Ни один из представленных вариантов не является верным.

6. Для чего предназначена диаграмма компонентов?

(?) Моделирования статической структуры классов системы и связей между ними.

(?) Моделирования бизнес-процессов организации и требований к создаваемой системе.

(?) Моделирования поведения объектов системы при переходе из одного состояния в другое.

(?) Моделирования физической архитектуры системы.

(!) Ни один из представленных вариантов не является верным.

7. Кем является действующее лицо в диаграмме вариантов использования?

- (?) Пользователем, который имеет право изменять модель.
- (?) Пользователем, имеющий право администратора в системе.
- (!) Роль, которую пользователь играет по отношению к системе.
- (?) Ни один из представленных вариантов не является верным.

8. Каким значком в диаграмме вариантов использования изображается действующее лицо?

- (?) Овалом.
- (?) Прямоугольником.
- (!) Человеческой фигуркой.
- (?) Прямоугольником со скругленными углами.

9. Что относится к диаграммам взаимодействия?

- (!) Диаграммы последовательности.
- (?) Диаграммы классов.
- (?) Диаграммы состояний.

10. Что содержат секции изображения класса, представленного в виде прямоугольника со сплошной границей, разделенного горизонтальными линиями на 3 секции?

- (?) Имя класса, методы, атрибуты.
- (!) Имя класса, атрибуты, методы.
- (?) Атрибуты, методы, отношения.
- (?) Методы, атрибуты, отношения.

11. Что такое авторская разработка?

- (?) У которой имеется автор.
- (!) Когда весь жизненный цикл разработки поддерживается одним человеком.

12. Что такое коллективная разработка?

- (?) Когда за результат отвечает весь коллектив.
- (!) Это разработка коллективом с распределением ролей и определенной иерархией подчиненности.

13. Что такое общинная разработка?

- (?) Когда коллектив разработчиков является общиной.
- (!) Это разработка с децентрализованным управлением и большим количеством тестеров.

14. Как устроена иерархическая модель коллектива разработчиков?

- (?) В ней все руководство проекта находится на одном (верхнем) уровне.
- (!) В этой структуре имеется менеджер проекта и специалисты по функциям.

15. Что такое модель MSF?

- (?) Это произвольная иерархическая структура разработчиков.
- (!) Это каркас структуры бригады разработчиков от компании Microsoft.

16. Какие основные ограничения следует соблюсти при выполнении проекта?

- (?) По численности коллектива.

(!) Стоимостные и временные.

17. Что такое спецификация?

(?) Это перечень издержек.

(!) Это подробное описание продукта, согласованное с заказчиком. В его основе лежит принцип «сделай все, что обещано».

18. Какова роль менеджера продукта в модели MSF?

(?) Контроль сроков разработки.

(!) Удовлетворение требований заказчика.

19. Какова роль менеджера программы в модели MSF?

(?) Определить основные временные характеристики разрабатываемого ПО.

(!) Соблюдение ограничений проекта.

20. Какова роль разработчика в модели MSF?

(?) Взаимодействие с заказчиком.

(!) Обеспечение выполнения спецификаций.

21. Какова роль тестера в модели MSF?

(?) Прогнозирование сроков разработки.

(!) Выпуск ПО только после выявления и устранения всех проблем.

22. Какова роль инструктора в модели MSF?

(?) Инструктирование новых членов коллектива разработчиков.

(!) Повышение производительности труда пользователя ПО.

23. Какова роль логистика в модели MSF?

(?) Обеспечение хранения резервных копий ПО.

(!) Простота развертывания и обеспечение сопровождения ПО.

24. Каков минимальный размер проектной группы?

(?) Шесть человек.

(!) Один человек.

25. Можно ли в небольших группах разработчиков совмещать разработку с другими видами деятельности?

(?) Да.

(!) Нет.

3.2 Примерная тематика контрольных работ, выполняемых с применением программных средств специального и общего назначения на компьютерах:

1. Выполнить коллективную разработку программы «Таксофон».

Программа должна быть реализована в настольной версии и иметь следующую логику работы.

Таксофон предназначен для оказания платных услуг телефонной связи. Он подключен к линии связи. В нем имеется кнопочная панель, дисплей, трубка со встроенным микрофоном и громкоговорителем, приемник карт – устройство для считывания телефонных карт, используемых для оплаты разговора.

В начальном состоянии трубка таксофона повешена, дисплей погашен, таксофон не реагирует на нажатия кнопок и какие-либо сигналы из линии. При снятии трубки таксофон выдает на дисплей сообщение «Вставьте карту» и

ожидает, когда пользователь вставит карту в приемник. Дальнейшее функционирование таксофона осуществляется только при вставленной карте. Если карту вынимают, таксофон возвращается к началу и выдает сообщение о необходимости вставить карту. При попадании карты в приемник производится считывание информации с карты. Если кредит исчерпан или карта не пригодна (не удастся узнать кредит), то таксофон выдает соответствующее сообщение на дисплей таксофона. Если карта может быть использована для оплаты, то на дисплей выдается количество «единиц» на карте, и на телефонную станцию (АТС) подается сигнал «Трубка». При получении ответного сигнала «Тон» из линии таксофон воспроизводит звуковой тон «Готов» (длинный непрерывающийся гудок) в трубку. При получении сигнала «Занято», в трубке воспроизводится тон «Занято» (короткие гудки).

После получения от АТС сигнала «Тон» от пользователя принимаются семизначный номер вызываемого абонента, остальные нажатия на кнопки игнорируются. Когда пользователь нажимает на кнопку с цифрой соответствующий ей сигнал «Цифра» передается АТС. Во время набора номера введенные цифры отображаются на дисплее. В ответ на набранный номер от АТС приходит либо сигнал «Занято», либо сигнал «Вызов». При получении сигнала «Вызов» таксофон воспроизводит в трубку длинные гудки до того момента, когда АТС осуществит коммутацию и передаст сигнал «Данные». Таксофон воспроизводит данные, передаваемые с сигналом, в трубку. При получении данных из трубки, аппарат преобразует их в сигнал «Данные» и передает их АТС. Во время разговора на дисплее ведется отсчет времени и уменьшается кредит на телефонной карте - каждые 15 секунд вычитается четверть «единицы». Обмен данными прерывается, в следующих случаях:

- исчерпан кредит;
- карта вынута из приемника;
- от АТС пришел сигнал «Занято»;
- повешена трубка таксофона.

Если трубка была повешена, аппарат посылает в линию сигнал «Конец» и выдает на дисплей сообщение «Выньте карту». После извлечения карты из приемника таксофон переходит в начальное состояние.

Во исполнение этого проекта следует собрать группу разработчиков и для выполнения проекта распределить между ними следующие роли:

- А.1. Заказчик.
- А.2. Менеджер продукта.
- А.3. Менеджер программы.
- А.4. Тестер.
- А.5. Инструктор.
- А.6. Логистик.
- А.7. Главный программист.
- А.8. Дублер главного программиста.

- А.9. Администратор.
- А.10. Технический редактор
- А.11. Языковед.
- А.12. Инструментальщик.
- А.13.Отладчик.
- А.14. Делопроизводитель

2. Выполнить коллективную разработку программы «Банкомат».

Программа должна быть реализована в настольной версии и иметь следующую логику работы.

Банкомат – это автомат для выдачи наличных денег по кредитным пластиковым карточкам. В его состав входят следующие устройства: дисплей, панель управления с кнопками, приемник кредитных карт, хранилище денег и лоток для их выдачи, хранилище конфискованных кредитных карт, принтер для печати справок.

Банкомат подключен к линии связи для обмена данных с банковским компьютером, хранящим сведения о счетах клиентов.

Обслуживание клиента начинается с момента помещения пластиковой карточки в банкомат. После распознавания типа пластиковой карточки, банкомат выдает на дисплей приглашение ввести персональный код. Персональный код представляет собой четырехзначное число. Затем банкомат проверяет правильность введенного кода. Если код указан неверно, пользователю предоставляются еще две попытки для ввода правильного кода. В случае повторных неудач карта перемещается в хранилище карт, и сеанс обслуживания заканчивается. После ввода правильного кода банкомат предлагает пользователю выбрать операцию. Клиент может либо снять наличные со счета, либо узнать остаток на его счету.

При снятии наличных со счета банкомат предлагает указать сумму (10, 50, 100, 200, 500, 1000 рублей). После выбора клиентом суммы банкомат запрашивает, нужно ли печатать справку по операции. Затем банкомат посылает запрос на снятие выбранной суммы центральному компьютеру банка. В случае получения разрешения на операцию, банкомат проверяет, имеется ли требуемая сумма в его хранилище денег. Если он может выдать деньги, то на дисплей выводится сообщение «Выньте карту». После удаления карточки из приемника, банкомат выдает указанную сумму в лоток выдачи. Банкомат печатает справку по произведенной операции, если она была затребована клиентом.

Если клиент хочет узнать остаток на счету, то банкомат посылает запрос центральному компьютеру банка и выводит сумму на дисплей. По требованию клиента печатается и выдается соответствующая справка.

В специальном отделении банкомата, закрываемом замком, есть «секретная кнопка», которая используется обслуживающим персоналом для загрузки денег. При нажатии на эту кнопку открывается доступ к хранилищу денег и

конфискованным кредитным картам.

Во исполнение этого проекта следует собрать группу разработчиков и для выполнения проекта распределить между ними следующие роли:

- В.1. Заказчик.
- В.2. Менеджер продукта.
- В.3. Менеджер программы.
- В.4. Тестер.
- В.5. Инструктор.
- В.6. Логистик.
- В.7. Главный программист.
- В.8. Дублер главного программиста.
- В.9. Администратор.
- В.10. Технический редактор
- В.11. Языковед.
- В.12. Инструментальщик.
- В.13. Отладчик.
- В.14. Делопроизводитель

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Формой контроля знаний по дисциплине являются два текущих контроля знаний в форме тестирования и зачет, проводимый по материалам лекций и выполненным практическим заданиям.

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающих знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
Проводится в сроки, установленные графиком образовательного процесса	Тестирование 1, 2	ПК-5, ПК-7	15 вопросов	Компьютерное тестирование; время, отведенное на процедуру - 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - 0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов
Проводится в сроки, установленные графиком образовательного процесса	Зачет	ПК-5, ПК-7	3 задания	Время, отведенное на процедуру – 45 минут.	Результаты предоставляются в день проведения зачета	Критерии оценки: «Зачтено»: • знание основных понятий предмета; • умение использовать и применять

го процесс а						полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • частичный ответ на вопросы билета «Не зачтено»: • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы.
--------------------	--	--	--	--	--	---

4.1. Типовые вопросы, выносимые на зачет

1. Какими характеристиками должен обладать любой программный продукт?
2. Каково различие между моделью процесса создания ПО и самим процессом?
3. Почему в процессе определения требований необходимо различать разработку пользовательских требований и разработку системных требований?
4. Каковы основные компоненты методов проектирования?
5. Почему процесс планирования проекта является итерационным и почему план должен постоянно пересматриваться в течение всего срока выполнения проекта.
6. Опишите кратко каждый раздел плана выполнения программного проекта.
7. Почему архитектурное проектирование системы должно предшествовать разработке формальной спецификации.
8. Почему архитектуру системы необходимо разработать до окончания создания спецификации.
9. Преимущества и недостатки модели потоков данных и объектной модели в предположении, что необходимо разработать как локальную, так и распределенную версии программного приложения.

10. 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.
11. Покажите на примерах разницу между объектом и классом объектов.
12. При каких условиях можно разрабатывать систему, в которой объекты выполняются параллельно?
13. Различия между верификацией и аттестацией и объясните, почему аттестация является более сложным процессом.
14. Почему не нужно устранять все дефекты в программе перед ее поставкой заказчику. До каких пор следует тестировать программу, чтобы удостовериться, что она соответствует своему назначению?
15. Почему инспектирование программы является эффективным методом обнаружения в ней ошибок. Какие типы ошибок нельзя обнаружить методом инспектирования?
16. Различия между тестированием методом черного ящика и структурным тестированием. Подумайте, каким образом можно совместно использовать эти методы в процессе тестирования дефектов.
17. Проблемы тестирования могут возникнуть в программах, которые обрабатывают как очень большие, так и очень малые числа?
18. Почему методы нисходящего и восходящего тестирования не подходят для объектно-ориентированных систем.
19. Подходы к определению производительности программиста. Преимущества и недостатки каждого подхода.
20. Факторы, которые оказывают существенное влияние на производительность команды программистов по разработке больших встроенных систем реального времени.
21. Способы снижения возможного риска при оценке стоимости.
22. Причины, по которым алгоритмические оценки стоимости, проведенные различными компаниями, не будут сопоставимыми.
23. Каким образом менеджеры проектов могут использовать алгоритмический подход к оценке стоимости для анализа проектных характеристик. Когда менеджеры выбирают подход, не основанный на принципе наименьшей стоимости проекта.
24. Какими основными принципами следует руководствоваться при использовании цветов в интерфейсах пользователя?
25. Почему системы реального времени обычно реализованы как множество параллельных процессов? Проиллюстрируйте свой ответ примерами.
26. Почему объектно-ориентированные методы разработки ПО не всегда подходят к системам реального времени.
27. Сильные и слабые стороны Java как языка программирования для реализации систем реального времени.
28. Кто должен проводить обзор требований? Нарисуйте модель процесса обзора требований.

Итоговое начисление баллов по дисциплине осуществляется в соответствии с разработанной и внедренной балльно-рейтинговой системой контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся.

*ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«ТЕХНОЛОГИИ И СИСТЕМЫ КОЛЛЕКТИВНОЙ РАЗРАБОТКИ
ПРОГРАММ»**

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Программирование, математическое моделирование

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев
2023

1. Общие положения

Цель дисциплины:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

Задачи дисциплины:

1. Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
2. Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

2. Указания по проведению практических занятий

Практическая работа 1.

Вид практического занятия: *компьютерное моделирование.*

Образовательные технологии: *технологии компьютерного обучения.*

Тема и содержание занятия: *Тема 1. Методология коллективной разработки ПО.*

- *Построение плана коллективной разработки проекта ПО в среде JIRA Software.*

Продолжительность занятия – 8 ч.

Практическая работа 2.

Вид практического занятия: *компьютерное моделирование.*

Образовательные технологии: *технологии компьютерного обучения.*

Тема и содержание занятия: *Тема 2. Планирование, мониторинг и управление работами коллектива разработчиков ПО.*

- *Загрузка / выгрузка проектной документации в системе Confluence.*

Продолжительность занятия – 12 ч.

Практическая работа 3.

Вид практического занятия: *компьютерное моделирование.*

Образовательные технологии: *технологии компьютерного обучения.*

Тема и содержание занятия: *Тема 3. Мониторинг версий и непрерывная интеграция проекта ПО.*

- *Мониторинг версий ПО в системах SubVersion или Git и интеграция в системе Jenkins.*

Продолжительность занятия – 12 ч.

3. Указания по проведению лабораторного практикума

Не предусмотрено учебным планом.

4. Указания по проведению самостоятельной работы обучающихся

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1.	Методология коллективной разработки ПО	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (алгоритм гибкой разработки).
2.	Планирование, мониторинг и управление работами коллектива разработчиков ПО	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (алгоритм распределения работ).
3.	Мониторинг версий и непрерывная интеграция проекта ПО	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (алгоритм автоматической сборки проекта).

5. Указания по проведению контрольных работ для обучающихся очной формы обучения

Пояснения по содержанию, выполнению и оформлению работы:

Контрольная работа проводится с использованием ЭВМ. Задача оценивается в 5 баллов. Учитывается наличие всех указанных в задаче объектов метаданных, процедур обработки информации в модулях форм или документов, удобного интерфейса.

ФОРМА ОТЧЕТНОСТИ

После создания проекта обучающиеся предоставляют письменный отчет о проделанной работе (контрольную работу).

Контрольная работа должна содержать:

Титульный лист установленного образца

Содержание

Индивидуальное задание (указать индивидуальное задание, цели и задачи выполнения данного задания)

1. Техника безопасности
2. Выбор средств и реализация поставленных задач (описание работы)
3. Выводы (самоанализ проделанной работы: итог, трудности, новое, интересное и т. д.)
4. Список литературы

Приложения

Критерии оценок:

«отлично» - все объекты метаданных присутствуют, обмен информацией верен, пользовательские процедуры написаны на встроенном языке и

выполняются без сбоев, интерфейс удобен, конфигурация успешно работает, созданы интерфейс и набор прав пользователей.

«хорошо» - все объекты метаданных присутствуют, обмен информацией может содержать незначительные ошибки, пользовательские процедуры написаны на встроенном языке и выполняются без сбоев, конфигурация успешно работает, созданы интерфейс и набор прав пользователей.

«удовлетворительно» - основные объекты метаданных присутствуют, обмен информацией может содержать незначительные ошибки, пользовательские процедуры на встроенном языке выполняются без сбоев, конфигурация успешно работает.

«неудовлетворительно» - основные объекты метаданных присутствуют, обмен информацией содержит ошибки, пользовательские процедуры на встроенном языке содержат ошибки либо вообще отсутствуют, конфигурация не работает.

Контрольная работа сдается на электронных носителях в виде каталога с названием фамилии студента, содержащего информационную базу и задачи на бумажном носителе в виде контрольной работы.

УКАЗАНИЯ ПО ОФОРМЛЕНИЮ ОТЧЕТА

Общие требования

Текстовые документы выполняются на листах белой бумаги стандартного формата А4 (Приложение В, Д). Для страницы документа устанавливаются поля: Верхнее - 1,6 мм; Левое – 2,4 мм; правое – 1,0 мм, нижнее – 3,1 мм.

Примечание: Параметры страницы устанавливаются в меню Файл->Параметры страницы:

Текстовые документы выполняются с применением персонального компьютера, в текстовом процессоре MS Word, шрифтом Times New Roman №16 (по усмотрению преподавателя может быть №14), междустрочный интервал 1,5, строчными буквами.

Поля устанавливаются для текста:

в начале строки – не менее 3 мм

в конце строки – не менее 3 мм.

Расстояние от верхней и нижней строки текста до линии рамки должно быть не менее 10 мм. (Приложение В, Г)

Поврежденные листы, помарки, следы прежнего, не полностью удаленного текста не допускаются.

Абзацный отступ – 1,25мм. Устанавливается в меню Формат->Абзац:

Построение документов

Текст документа при необходимости разделяют на разделы и подразделы.

Разделы должны иметь порядковые номера, обозначенные арабскими цифрами с точкой в пределах документа. Подразделы должны иметь нумерацию в пределах раздела. Номера подразделов состоят из номера раздела и подраздела, разделенной точкой.

В конце номера подраздела, также ставится точка.

Разделы, как и подразделы, могут состоять из одного или нескольких пунктов

Если раздел состоит из одного пункта, он нумеруется. Номер пункта состоит из номера раздела, номера подраздела и номера пункта, разделенных точками.

Подпункты в пределах пункта или перечисление требований, указаний, положений обозначаются арабскими цифрами со скобкой, например, 1), 2), 3) и т.д.

Каждый пункт, подпункт, перечисления записывают с нового абзаца, шрифтом Times New Roman № 18

Наименование разделов и подразделов записываются в виде заголовка, выровненного по центру, шрифтом Times New Roman №22, прописными буквами.

Наименование разделов и подразделов записывается в виде заголовка, шрифтом Times New Roman №22, строчными буквами, первая буква прописная.

В конце заголовка точку не ставят.

Расстояние между заголовками раздела и подразделов 10 мм (межстрочный интервал полуторный), между заголовками и текстом 15 мм (межстрочный интервал двойной), между последней строкой текста и последующим заголовком должно быть 15мм. (межстрочный интервал двойной).

Каждый раздел документа следует начинать с нового листа.

Каждый пункт текста записывается с абзаца (отступ красной строки 15 мм.), цифры, указывающие номер пунктов, не должны выступать за границу абзаца.

Оформление формул

В формулах в качестве символов следует применять обозначения, установленные соответствующими стандартами

Если в документе более одной формулы, их нумеруют арабскими цифрами в пределах раздела или всего документа. Номер указывают с правой стороны листа, арабскими цифрами на уровне формулы в круглых скобках.

Оформление иллюстраций и приложений

Количество иллюстраций должно быть достаточным для пояснения излагаемого текста.

Все иллюстрации нумеруются в пределах всего документа арабскими цифрами сквозной нумерации, за исключением иллюстраций приложений.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения.

6. Перечень основной и дополнительной учебной литературы

Основная литература:

1. Соснин П. И. Человеко-компьютерное взаимодействие: учебное пособие / П. И. Соснин, В. В. Валюх. — Ульяновск: УлГТУ, 2020. — 119 с. — ISBN 978-5-9795-2020-9. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/259775>
2. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.: КУРС, НИЦ ИНФРА-М, 2016. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6, URL: <http://znanium.com/catalog/product/551224>.

Дополнительная литература:

1. Интеллектуальные системы проектирования и управления техническими объектами: учебное электронное издание: учебное пособие: в 4 частях / В. А. Немтинов, С. В. Карпушкин, В. Г. Мокрозуб [и др.]; Тамбовский государственный технический университет. – Тамбов: Тамбовский государственный технический университет (ТГТУ), 2018. – Часть 3. – 153 с.: табл., граф., схем., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=570332>
2. Арыков, С.Б. Параллельное программирование над общей памятью: OpenMP: [16+] / С.Б. Арыков, М.А. Городничев, Г.А. Щукин; Новосибирский государственный технический университет. – Новосибирск: Новосибирский государственный технический университет, 2019. – 95 с.: ил., табл. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=576119>.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

- <http://www.znanium.com/> - электронно-библиотечная система
- <http://www.e.lanbook.com/> - ЭБС Издательства "ЛАНЬ"
- <http://www.rucont.ru/> - электронно-библиотечная система
- <http://www.biblioclub.ru/> - университетская библиотека онлайн.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения: MSOffice

Информационные справочные системы: Электронные ресурсы образовательной среды Университета