**Программа учебной дисциплины**

**Научно-исследовательский семинар
"Программная инженерия: технологии программирования"**

**Первый год обучения**

Утверждена

Академическим советом ООП

Протокол № от «\_\_»\_\_\_\_\_20\_\_ г.

|  |  |
| --- | --- |
| Автор  | А.И.Аветисян, А.К.Петренко |
| Число кредитов  | 8 |
| Контактная работа (час.)  | 96 |
| Самостоятельная работа (час.)  | 208 |
| Курс  | 1 |
| Формат изучения дисциплины | full time |

1. **ЦЕЛЬ, РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ И ПРЕРЕКВИЗИТЫ**

Целями освоения дисциплины "Программная инженерия: технологии программирования" являются:

* обеспечить студентов базовыми знаниями в области системного программирования;
* заложить основы для последующих курсов, посвященных созданию современных средств системного программирования;
* обучить студентов применению современных интегрированных инструментальных средств, предназначенных для разработки системного программного обеспечения (ПО);
* привить студентам навыки исследовательской работы, предполагающей самостоятельное изучение специфических инструментов и средств, необходимых для решения именно той конкретной проблемы, которая в качестве задачи поставлена перед ними.

В результате освоения дисциплины студент должен:

**Знать:**

* + основные языки системного программирования;
	+ основные принципы архитектурного дизайна системного ПО;
	+ основные программные платформы.

**Уметь:**

* + разрабатывать системные программы с помощью инструментальных средств;
	+ отлаживать и тестировать создаваемые программы, используя диагностические возможности среды разработки;
	+ самостоятельно находить новые знания и решения, необходимые для реализации функциональных требований, сформулированных в техническом задании на программу.

**Владеть навыками (приобрести опыт):**

* + в решении типовых задач системного программирования с применением современных языков программирования и передовых инструментальных средств;
	+ проектирования и программирования системных программ;
	+ разрабатывать требования верхнего уровня и детальные требований к компонентами программных систем;
	+ в применении библиотек используемых платформ и свободно распространяемых библиотек.

Изучение дисциплины "Программная инженерия: технологии программирования" базируется на следующих дисциплинах:

* «Дискретная математика»,
* «Программирование»,
* «Построение и анализ алгоритмов»,
* «Архитектура вычислительных систем»,
* «Операционные системы».

Для освоения учебной дисциплины студенты должны владеть следующими знаниями и компетенциями:

* уметь программировать на распространенных языках программирования;
* уметь использовать операционные системы типа Unix/Linux;
* уметь работать с СУБД.
1. **СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**

**Тема 1. Предмет и методы системного программирования.**

Понятие системного слоя программного обеспечения. Особенности разработки и эксплуатации системного ПО. Виды системного ПО и виды инструментов разработки и анализа системного ПО.

**Вычислительные машины.**

Многопроцессорные и многомашинные вычислительные системы. СуперЭВМ. Локальные и глобальные сети ЭВМ. Базовая эталонная модель взаимодействия открытых систем

**Тема 2. Инструменты поддержки жизненного цикла ПО. Управление требованиями. Методы проектирования программ на основе моделей.**

Виды моделей и парадигмы моделирования. Исполнимые (явные) модели, модели ограничений, алгебраические модели.

Принципы системного проектирования комплексов программ. Cистемная и программная инженерия, процессы жизненного цикла сложных технических систем и программных комплексов.

Декомпозиция требований, функций, процессов проектирования компонентов и комплексов программ. Повторное использование готовых компонентов при проектировании программных комплексов.

**Тема 3. Методы проектирования программ на основе моделей. Методы верификации. Тестирование на основе моделей.**

Виды моделей и парадигмы моделирования. Исполнимые (явные) модели, модели ограничений, алгебраические модели. Подходы к разработке и верификации программ на основе моделей (MDA, MDSE, MBT, SBT).

Процессы верификации компонентов и комплексов программ. Трассирование взаимодействия требований к компонентам в комплексах программ. Организация процессов тестирования компонентов и комплексов программ. Процессы и методы тестирования программных модулей и компонентов. Функциональная декомпозиция системы тестирования на основе моделей. Примеры известных приложений подхода к тестированию на основе моделей.

**Тема 4. Языки и компиляторы.**

Языки и грамматики. Синтаксис и семантика языков программирования. Формальное определение грамматики и языка. Ассемблеры и загрузчики. Команды и псевдокоманды, символические адреса, адресные выражения. Характерные особенности языков программирования: C, C++, Java, C#, Python. Основные этапы работы компилятора: лексический анализ, синтаксический анализ и генерация промежуточного кода, генерация объектного кода, оптимизация кода. Промежуточное представление. Польская запись, тетрады, триады и деревья. Оптимизация программ. Основные методы оптимизации. Машиннозависимая и машиннонезависимая оптимизации. Генерация объектного кода. Нейтрализация семантических и синтаксических ошибок.

**Тема 5. Операционные системы.**

Функции и основные понятия операционной системы. Процессы. Реализация процессов. Взаимодействие процессов. Параллельные процессы. Взаимное исключение и взаимная синхронизация. Примитивы синхронизации: события, семафоры, мониторы Хоара, почтовые ящики. Распределение времени процессора. Мультипрограммирование. Методы планирования в мультипрограммных системах. Управление памятью. Распределение памяти и организация доступа к памяти в ЭВМ с различной структурой памяти. Виртуальная память. Стратегии и методы замещения страниц. Ввод-вывод в файлы. Базисная и логическая системы управления файлами. Методы доступа к файлам. Планирование заданий.

**Тема 6. Системы управления базами данных.**

Физическая организация данных. Перемешанные файлы. Индексированные файлы. В-деревья. Модели баз данных и их особенности: реляционная, сетевая и иерархическая модели. Управление данными в реляционной модели. Реляционная алгебра. Реляционное исчисление. Защита базы данных. Целостность. Секретность. Организация мультидоступа к базе данных. Транзакция. Синхрозахваты.

1. **ОЦЕНИВАНИЕ**

В ходе работ научно-исследовательского семинара студенты готовят и представляют доклады по темам занятий, участвуют в их обсуждении. Оценки выставляются за качество подготовленных докладов, за активное и содержательное обсуждение докладов других участников семинара и за ответы на экзамене.

Преподаватель оценивает работу студентов на семинарских занятиях: Оценки за работу на семинарских занятиях преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за работу на семинарских занятиях определяется перед итоговым контролем - *Оауд*.

Преподаватель оценивает самостоятельную работу студентов. Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость. Накопленная оценка по 10-ти балльной шкале за самостоятельную работу определяется перед итоговым контролем – *Осам. работа*.

Накопленная оценка за текущий контроль учитывает результаты студента по текущему контролю следующим образом:

О*накопленная*= *0,5*\* Оауд + *0,5*\*Осам.работа

На пересдаче студенту не предоставляется возможность получить дополнительный балл для компенсации оценки за текущий контроль.

На экзамене студент может получить дополнительный вопрос, ответ на который оценивается в 1 балл.

В диплом выставляет результирующая оценка по учебной дисциплине, которая формируется по следующей формуле:

*Орезульт = 0,5*\**Онакопл* + *0,5*\**Оэкз*

1. **ПРИМЕРЫ ОЦЕНОЧНЫХ СРЕДСТВ**

# Примеры тем докладов:

* Виды операционных систем и их особенности.
* Жизненный цикл разработки программ и инструменты его поддержки.
* Методы проектирования программ, виды моделирования программ.

Оценочные средства для итоговой аттестации (варианты вопросов на экзамене)

* Методы машиннозависимой оптимизации программ.
* Стратегии планирования в операционных системах.
* Методы управления требованиями в программных проектах.
1. **РЕСУРСЫ**
	1. **Основная литература**
2. В.В.Кулямин, Н.В.Пакулин, О.Л.Петренко, А.А.Сортов, А.В.Хорошилов. [Формализация требований на практике.](http://www.ispras.ru/publications/2006/formalizatsiya_trebovaniy_na_praktike/) Препринт Института системного программирования РАН, №13, 2006.
3. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. Вильямс, 2010.
4. Ахо Альфред В., Хопкрофт Джон Ульман, Джеффри, Д. Структуры данных и алгоритмы: Пер. с англ. : Уч. пос. - М. : Издательский дом "Вильямс", 2000.
5. Стивенс Р., Раго С. UNIX. Профессиональное программирование. - СПб.: Символ-Плюс, 2007.
6. Молчанов А.Ю. Системное программное обеспечение: Учебник для вузов. 3-е изд., "Питер", 2009, 400 с.
7. В.В.Кулямин, Н.В.Пакулин, О.Л.Петренко, А.А.Сортов, А.В.Хорошилов. [Формализация требований на практике.](http://www.ispras.ru/publications/2006/formalizatsiya_trebovaniy_na_praktike/) Препринт Института системного программирования РАН, №13, 2006.
8. В.В. Кулямин. [Технологии программирования. Компонентный подход.](http://www.ispras.ru/publications/2007/tekhnologii_programmirovaniya_komponentnyy_podkhod/) М. Интернет-университет информационных технологий - БИНОМ. Лаборатория знаний, 2007.
9. В. В. Кулямин. [Методы верификации программного обеспечения.](http://www.ispras.ru/publications/2008/methods_of_software_verification/) Конкурс обзорно-аналитических статей по направлению «Информационно-телекоммуникационные системы», 2008.
10. Липаев В.В. Программная инженерия сложных заказных программных продуктов // Учебное пособие. – М.: МАКС Пресс, 2014. – 312 с. (<http://www.ispras.ru/lipaev/books/Software%20Engineering%20of%20Complex%20Custom%20Software.pdf> и <http://www.directmedia.ru/book_278971_programmnaya_injeneriya_slojnyih_zakaznyih_programmnyih_produktov/>).
11. K.R. Apt, F.S. de Boer, E.-R. Olderog. Verification of Sequential and Concurrent Programs, Springer, 2009.
12. C. Baier, J.-P. Katoen. Principles of Model Checking. The MIT Press, 2008.
13. Ю.Г. Карпов. Model Checking. Верификация параллельных и распределенных программ- ных систем. БХВ-Петербург, 2010.
14. P. Baudin, P. Cuoq, J.-C. Filliâtre, C. Marché, B. Monate, Y. Moy, V. Prevosto. ACSL: ANSI/ISO C Specification Language. Version 1.4. 2010.
15. G.J. Holzmann. The SPIN Model Checker. Primer and Reference Manual. Addison-Wesley, 2003.
16. Э.М. Кларк, О. Грамберг, Д. Пелед. Верификация моделей программ. Model Checking. М.: МЦНМО, 2002.
17. Э. Дейкстра. Дисциплина программирования. М.: Мир, 1978.

**Дополнительная литература**

1. Александр Петренко, Елена Бритвина, Сергей Грошев, Александр Монахов, Ольга Петренко. [Тестирование на основе моделей](http://citforum.ru/SE/testing/model/) // [Открытые системы, #09/2003](http://www.osp.ru/os/2003/09/) (<http://citforum.ru/SE/testing/model/>).
2. Липаев В.В. Программная инженерия сложных заказных программных продуктов // Учебное пособие. – М.: МАКС Пресс, 2014. – 312 с. (<http://www.ispras.ru/lipaev/books/Software%20Engineering%20of%20Complex%20Custom%20Software.pdf> и <http://www.directmedia.ru/book_278971_programmnaya_injeneriya_slojnyih_zakaznyih_programmnyih_produktov/>).
	1. **Программное обеспечение (не требуется)**
	2. **Профессиональные базы данных, информационные справочные системы,
	интернет-ресурсы (электронные образовательные ресурсы)**

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Наименование** | **Условия доступа** |
|  | ***Профессиональные базы данных, информационно-справочные системы*** |
| 1. | Электронно-библиотечная система Юрайт  | URL: https://biblio-online.ru/ |
|  | ***Интернет-ресурсы (электронные образовательные ресурсы)*** |
| 2. | Открытое образование  | URL: https://openedu.ru/ |

* 1. **Материально-техническое обеспечение дисциплины**

Учебные аудитории для лекционных занятий по дисциплине обеспечивают использование и демонстрацию тематических иллюстраций, соответствующих программе дисциплины в составе:

ПЭВМ с доступом в Интернет (операционная система, офисные программы, антивирусные программы);

мультимедийный проектор с дистанционным управлением.