

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Саровский физико-технический институт -
филиал федерального государственного автономного образовательного учреждения
высшего образования «Национальный исследовательский ядерный университет «МИФИ»
(СарФТИ НИЯУ МИФИ)

ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра «Цифровых технологий»

УТВЕРЖДАЮ

Декан ФТФ, член корр. РАН, д.ф-м.н.

_____ **А.К. Чернышев**

« ____ » _____ **2023 г.**

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ПРОЕКТИРОВАНИЕ ИНТЕРФЕЙСОВ

наименование дисциплины

Направление подготовки (специальность)	09.03.02 Информационные системы и технологии
Наименование образовательной программы	Инновационные технологии комплексной автоматизации и сквозного управления жизненным циклом
Квалификация (степень) выпускника	бакалавр
Форма обучения	очная

Программа одобрена на заседании кафедры

Зав. кафедрой ЦТ

О.В. Кривошеев

протокол № _____ от _____ **2023 г.**

« ____ » _____ **2023 г.**

г. Саров, 2023 г.

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ЦТ

О.В. Кривошеев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ЦТ

О.В. Кривошеев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ЦТ

О.В. Кривошеев

Программа переутверждена на 202____/202____ учебный год с изменениями в соответствии с семестровыми учебными планами академических групп ФТФ, ФИТЭ на 202____/202____ учебный год.

Заведующий кафедрой ЦТ

О.В. Кривошеев

Семестр	В форме практической подготовки	Трудоемкость, кред.	Общий объем курса, час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	КР/ КП	Форма(ы) контроля, экз./зач./ЗсО/	Интерактивные часы
7	32	4	144	-	-	32	76	-	Э	10
ИТОГО	32	4	144	-	-	32	76	-	36	10

АННОТАЦИЯ

Дисциплина «Проектирование интерфейсов» обеспечивает не только нормативно-методическую базу освоения обучающимися профессиональных компетенций в соответствии с требованиями ОС ВО по направлению подготовки 09.03.02 «Информационные системы и технологии», с квалификацией выпускника бакалавр, но и высокую профессиональную конкурентоспособность выпускников и их востребованность для решения актуальных задач у потребностей регионального и Всероссийского рынка труда, с учетом перспектив его развития.

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Целью освоения учебной дисциплины «Проектирование интерфейсов» является развитие у студентов знаний, умений, практических навыков и компетенций, которые позволяют:

- определять области применения GNU GPL, ее преимущества и недостатки, возможности применения при разработке программных средств, а также разрабатывать программные средства на основе GNU GPL;
- подготовить их к самостоятельной научно-исследовательской деятельности, выработке организованности, трудолюбия, коммуникабельности, способностей к быстрому и самостоятельному приобретению новых знаний.

Цель достигается решением поставленных задач в рамках выполнения студентами практических и самостоятельных работ с использованием методических разработок и контроля выполнения работ преподавателем.

2. МЕСТО УЧЕБНОЙ ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП ВО

Дисциплина «Проектирование интерфейсов» входит в вариативную часть рабочего учебного плана подготовки бакалавров по профилю «Инновационные технологии комплексной автоматизации и сквозного управления жизненным циклом» направления 09.03.02 «Информационные системы и технологии» и является выборочной.

Изучение учебной дисциплины «Проектирование интерфейсов» осуществляется после изучения дисциплин «Программное обеспечение с открытым кодом», «Методы и средства проектирования информационных систем» и технологий, а также является логическим продолжением дисциплины «Операционные системы GNU и BSD».

Базовые «входные» знания, которыми должен обладать студент перед изучением учебной дисциплины «Проектирование интерфейсов», представляют собой знание основ разработки

программных средств, знание языка программирования, а также базовых алгоритмов и структур данных.

3. ФОРМИРУЕМЫЕ КОМПЕТЕНЦИИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Ожидается, что в результате освоения дисциплины студент приобретет следующие компетенции:

Профессиональные компетенции (ПК)

в соответствии с задачами и объектами (областями знаний) профессиональной деятельности:

Задача профессиональной деятельности (ЗПД)	Объект или область знания	Код и наименование профессиональной компетенции	Код и наименование индикатора достижения профессиональной компетенции
Тип задачи профессиональной деятельности: проектный			
проектирование базовых и прикладных информационных технологий	информационные процессы, технологии, системы и сети, их инструментальное (программное, техническое, организационное) обеспечение, способы и методы проектирования, отладки, производства и эксплуатации информационных технологий и систем в различных областях и сферах деятельности.	ПК-11 Способен проводить анализ предметной области и предпроектное обследование объекта проектирования с использованием формальных методов системного подхода <i>Основание:</i> Профессиональный стандарт «06.015. Специалист по информационным системам».	З-ПК-11 Знать: основные принципы системного подхода; этапы предпроектного обследования объекта проектирования У-ПК-11 Уметь: проводить анализ предметной области и предпроектное обследование объекта проектирования В-ПК-11 Владеть: инструментальными средствами описания предметной области.

		<p>ПК-12 Способен проводить выбор исходных данных для проектирования с учетом требований заказчика</p> <p><i>Основание:</i> Профессиональный стандарт «06.016. Руководитель проектов в области информационных технологий»</p>	<p>З-ПК-12 Знать: требования к разработке технического задания, его структуру и принципы составления</p> <p>У-ПК-12 Уметь: анализировать исходную документацию заказчика</p> <p>В-ПК-12 Владеть: методикой составления технического задания</p>
--	--	--	--

4. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ*

№ п/п	Наименование раздела/темы дисциплины	№ недели	Виды учебной работы					
			Лекции	Практ. занятия/ семинары	Лаб. работы	СРС	Текущий контроль (форма)*	Максимальн ый балл
			-		32	76		
Семестр 7								
Раздел 1. Знакомство с IDE и ЯП								
1.1.	Тема 1. Знакомство со средой разработки JetBrains PyCharm	1-2			8	12	УО	5
1.2	Тема 2. Введение в ЯП Python	3-4			8	12	УО	5
Рубежный контроль		4	ИКЗ					10
Раздел 2. Создание проекта командой разработчиков								
2.1	Тема 1. Постановка задачи проекта	5-6			4	12	УО	5
2.2	Тема 2. Создание макета приложения	7-8			4	10	КР	5
2.3	Тема 3. Разработка модулей приложения	9-12			4	16	КР	5
2.4	Тема 4. Сборка приложения	13-15			4	14	КР	5
Рубежный контроль		16	ИКЗ					10
Промежуточная аттестация			Экзамен				36	45
Посещаемость								5
Итого:			-		32	76	36	100

*Сокращение наименований форм текущего, рубежного и промежуточного контроля:

УО – устный опрос

КР – контрольная работа

ИКЗ – индивидуальное контрольное задание

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
7 семестр		
Не предусмотрен		

Лабораторные занятия

№	Наименование раздела /темы дисциплины	Содержание
7 семестр		
Раздел 1. Знакомство с IDE и ЯП		
1.1	Тема 1. Знакомство со средой разработки JetBrains PyCharm	<p>Данная тема посвящена обзору IDE PyCharm от компании JetBrains. Студенты познакомятся с основным перечнем задач, которые могут возникнуть в разработчика:</p> <ul style="list-style-type: none"> - подготовка к установке и установка IDE на рабочей станции - первичная настройка IDE PyCharm - обзор основных функциональных возможностей данной среды разработки - обзор часто используемых библиотек, в том числе и под лицензией GNU GPL / BSD.
1.2	Тема 2. Введение в ЯП Python	<p>На данном этапе обучения студенты познакомятся с языком программирования Python. Основная задача данной темы:</p> <ul style="list-style-type: none"> - подготовка к установке и установка ЯП Python на рабочей станции - обзор основных возможностей и создание проекта «Hello World» средствами IDE PyCharm
Раздел 2. Создание проекта командой разработчиков		
2.1	Тема 1. Постановка задачи проекта	<p>Практическое применение средств разработки GNU GPL подразумевает использование этих средств разработки для создания собственного проекта в целях закрепления получаемых навыков и знаний. Поэтому группой студентов будет разработан проект, разбитый на подпроекты-модули, каждый из которых будет вести один или несколько человек. Через это первым этапом при разработке ПО следует постановка задачи.</p>
2.2	Тема 2. Создание макета приложения	<p>После того, как задача была поставлена следует создание макета приложения: разработка концепции, разбивка приложения на функциональные модули. Описание модулей, закрепление за каждым модулем одного или нескольких разработчиков.</p>
2.3	Тема 3. Разработка модулей приложения	<p>Самый времязатратный этап. Группа студентов-разработчиков в течении нескольких лабораторных пар будет вести параллельную разработку нескольких функциональных модулей приложения,</p>

		используя средства разработки GNU GPL (различные библиотеки и модули ЯП Python), которые были ими изучены ранее.
2.4	Тема 4. Сборка приложения	<p>Заключительный этап, в котором все полученные знания и опыт инкапсулируются в виде законченного проекта. Задача этой темы – научить студентов совмещать отдельно разработанные модули в законченное решение согласно разработанному макету.</p> <p>Итогом курса является действующее ПО с необходимым перечнем функциональных возможностей.</p>

4.3. Перечень учебно-методического обеспечения для самостоятельной работы студентов

1. Операционная система GNU и движение за свободные программы <https://www.gnu.org/>
2. Документация проекта GNU <https://www.gnu.org/doc/other-free-books.html>
3. GNU LINUX — подробный обзор операционной системы <https://linuxvsem.ru/about-linux/gnu-linux-obzor>
4. Каталог свободных программ https://directory.fsf.org/wiki/Main_Page
5. Обзор открытых лицензий https://www.opennet.ru/base/sys/open_licenses.txt.html
6. <https://www.jetbrains.com/ru-ru/pycharm/>
7. <https://docs.python.org/3/>
8. <https://stackoverflow.com/>

Самостоятельная работа направлена на закрепление и углубление полученных теоретических и практических знаний. Включает в себя:

- ✓ работу с предыдущим лекционным материалом;
- ✓ самостоятельное изучение отдельных тем дисциплины;
- ✓ поиск и обзор литературы и электронных источников;
- ✓ чтение и изучение учебника и учебных пособий.

5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Фонд оценочных средств по дисциплине обеспечивает проверку освоения планируемых результатов обучения (компетенций и их индикаторов) посредством мероприятий текущего, рубежного и промежуточного контроля по дисциплине.

5.1. Паспорт фонда оценочных средств по дисциплине

Связь между формируемыми компетенциями и формами контроля их освоения представлена в следующей таблице:

Раздел	Темы занятий	Компетенция	Индикаторы освоения	Текущий контроль, неделя
Семестр 3				
Раздел 1	Тема 1. Знакомство со средой разработки JetBrains PyCharm	ПК-11 ПК-12	3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	УО 1-2
	Тема 2. Введение в ЯП Python		3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	УО 3-4
Рубежный контроль		ПК-11 ПК-12	3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	ИКЗ 4
Раздел 2	Тема 1. Постановка задачи проекта	ПК-11 ПК-12	3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	УО 5-6
	Тема 2. Создание макета приложения		3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	КР 7-8
	Тема 3. Разработка модулей приложения		3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	КР 9-12
	Тема 4. Сборка приложения		3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	КР 13-15
Рубежный контроль		ПК-11 ПК-12	3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	ИКЗ 16
Промежуточная аттестация		ПК-11 ПК-12	3-ПК-11; У-ПК-11; В-ПК-11 3-ПК-12; У-ПК-12; В-ПК-12	Экзамен

5.2. Примерные контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

5.2.1. Оценочные средства для текущего контроля

5.2.1.1. Примерные вопросы для устного опроса (УО)

1. Подразумевает ли выражение “свободная программа” применение GPL?
2. Все ли программы GNU используют в качестве лицензии GNU GPL?
3. Можно изменить GPL и сделать другую лицензию?
4. Почему GPL разрешает пользователям публиковать свои измененные версии?
5. Позволяет ли GPL продавать копии программы за деньги?
6. Позволяет ли GPL распространять копии по договору о неразглашении?
7. Почему GPL требует передачи копии GPL в каждой копии программы?
8. Требуется ли заявление авторских прав на изменения в программе под GPL?
9. Что GPL говорит о переводе программы на другой язык программирования?
10. Что имеют в виду, когда говорят, что две лицензии “совместимы”?

11. Что означают слова “лицензия совместима с GPL”?
12. Почему первоначальная лицензия BSD несовместима с GPL?
13. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
14. Можно ли воспользоваться GPL как лицензией для аппаратуры?
15. Как LGPL работает с Java?
16. В GPLv3 “предоставление в общий доступ” приводится как пример перемещения. Что это значит? Является ли такое предоставление формой передачи?
17. Что такое тивоизация? Как GPLv3 ее предотвращает?
18. Запрещает ли GPLv3 цифровое управление ограничениями?
19. Есть ли в GPLv3 “параграф о защите от патентных исков”?
20. Что означает слово “данная программа” в GPLv3? Означает ли оно всякую программу, когда-либо выпущенную под GPLv3?
21. Как обновить (L)GPLv2 до (L)GPLv3?
22. Почему GPL требует передачи копии GPL в каждой копии программы?
23. Почему нужно помещать уведомление о лицензии в каждый файл исходного текста?
24. Может ли разработчик программы, распространявший ее по GPL, впоследствии передать исключительные права на нее другой стороне?
25. Есть ли способ распространить GPL на выходные данные, которые люди получают при использовании моей программой?
26. Как GPL сочетается со шрифтами?
27. Можно ли выпустить измененную версию программы под GPL только в двоичном виде?
28. Можно ли выпускать программу под лицензией, в которой сказано, что ее измененные версии распространять под GPL можно, а саму первоначальную версию — нельзя?
29. Является ли размножение и применение многих копий внутри одной организации или предприятия “распространением”?
30. Является ли распространением передача копии в контролируемое дочернее предприятие?
31. Можно ли в программе установки просить людей нажать кнопку в знак согласия с GPL?
32. Может кто-нибудь, кто передает программы под GPLv3 в пользовательском продукте, применять удаленную аттестацию для предотвращения модификации этих программ пользователем?
33. Что в GPLv3 означает выражение “правила и протоколы связи по сети”?
34. В каких случаях GPL распространяется и на результат работы программы под GPL?
35. Совместима ли GPLv3 с GPLv2?
36. Насколько различные лицензии GNU совместимы друг с другом?

37. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
38. Когда программа и ее внешние модули считаются единой комбинированной программой?
39. Можно ли применять GPL, когда пишется внешний модуль к несвободной программе?
40. Какие юридические проблемы возникают, если применяются несовместимые с GPL библиотеки с программами под GPL?

5.2.1.2. Примерные вопросы контрольной работы (КР)

1. Подразумевает ли выражение “свободная программа” применение GPL?
2. Все ли программы GNU используют в качестве лицензии GNU GPL?
3. Можно изменить GPL и сделать другую лицензию?
4. Почему GPL разрешает пользователям публиковать свои измененные версии?
5. Позволяет ли GPL продавать копии программы за деньги?
6. Позволяет ли GPL распространять копии по договору о неразглашении?
7. Почему GPL требует передачи копии GPL в каждой копии программы?
8. Требуется ли заявление авторских прав на изменения в программе под GPL?
9. Что GPL говорит о переводе программы на другой язык программирования?
10. Что имеют в виду, когда говорят, что две лицензии “совместимы”?
11. Что означают слова “лицензия совместима с GPL”?
12. Почему первоначальная лицензия BSD несовместима с GPL?
13. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
14. Можно ли воспользоваться GPL как лицензией для аппаратуры?
15. Как LGPL работает с Java?
16. В GPLv3 “предоставление в общий доступ” приводится как пример перемещения. Что это значит? Является ли такое предоставление формой передачи?
17. Что такое тивоизация? Как GPLv3 ее предотвращает?
18. Запрещает ли GPLv3 цифровое управление ограничениями?
19. Есть ли в GPLv3 “параграф о защите от патентных исков”?
20. Что означает слово “данная программа” в GPLv3? Означает ли оно всякую программу, когда-либо выпущенную под GPLv3?
21. Как обновить (L)GPLv2 до (L)GPLv3?
22. Почему GPL требует передачи копии GPL в каждой копии программы?
23. Почему нужно помещать уведомление о лицензии в каждый файл исходного текста?

24. Может ли разработчик программы, распространявший ее по GPL, впоследствии передать исключительные права на нее другой стороне?
25. Есть ли способ распространить GPL на выходные данные, которые люди получают при использовании моей программой?
26. Как GPL сочетается со шрифтами?
27. Можно ли выпустить измененную версию программы под GPL только в двоичном виде?
28. Можно ли выпускать программу под лицензией, в которой сказано, что ее измененные версии распространять под GPL можно, а саму первоначальную версию — нельзя?
29. Является ли размножение и применение многих копий внутри одной организации или предприятия “распространением”?
30. Является ли распространением передача копии в контролируемое дочернее предприятие?
31. Можно ли в программе установки просить людей нажать кнопку в знак согласия с GPL?
32. Может кто-нибудь, кто передает программы под GPLv3 в пользовательском продукте, применять удаленную аттестацию для предотвращения модификации этих программ пользователем?
33. Что в GPLv3 означает выражение “правила и протоколы связи по сети”?
34. В каких случаях GPL распространяется и на результат работы программы под GPL?
35. Совместима ли GPLv3 с GPLv2?
36. Насколько различные лицензии GNU совместимы друг с другом?
37. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
38. Когда программа и ее внешние модули считаются единой комбинированной программой?
39. Можно ли применять GPL, когда пишется внешний модуль к несвободной программе?
40. Какие юридические проблемы возникают, если применяются несовместимые с GPL библиотеки с программами под GPL?

5.2.2 Оценочные средства для рубежного контроля

5.2.2.1 Примерные вопросы для индивидуального контрольного задания (ИКЗ)

1. Подразумевает ли выражение “свободная программа” применение GPL?
2. Все ли программы GNU используют в качестве лицензии GNU GPL?
3. Можно изменить GPL и сделать другую лицензию?
4. Почему GPL разрешает пользователям публиковать свои измененные версии?
5. Позволяет ли GPL продавать копии программы за деньги?
6. Позволяет ли GPL распространять копии по договору о неразглашении?
7. Почему GPL требует передачи копии GPL в каждой копии программы?

8. Требуется ли заявление авторских прав на изменения в программе под GPL?
9. Что GPL говорит о переводе программы на другой язык программирования?
10. Что имеют в виду, когда говорят, что две лицензии “совместимы”?
11. Что означают слова “лицензия совместима с GPL”?
12. Почему первоначальная лицензия BSD несовместима с GPL?
13. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
14. Можно ли воспользоваться GPL как лицензией для аппаратуры?
15. Как LGPL работает с Java?
16. В GPLv3 “предоставление в общий доступ” приводится как пример перемещения. Что это значит? Является ли такое предоставление формой передачи?
17. Что такое тивоизация? Как GPLv3 ее предотвращает?
18. Запрещает ли GPLv3 цифровое управление ограничениями?
19. Есть ли в GPLv3 “параграф о защите от патентных исков”?
20. Что означает слово “данная программа” в GPLv3? Означает ли оно всякую программу, когда-либо выпущенную под GPLv3?
21. Как обновить (L)GPLv2 до (L)GPLv3?
22. Почему GPL требует передачи копии GPL в каждой копии программы?
23. Почему нужно помещать уведомление о лицензии в каждый файл исходного текста?
24. Может ли разработчик программы, распространявший ее по GPL, впоследствии передать исключительные права на нее другой стороне?
25. Есть ли способ распространить GPL на выходные данные, которые люди получают при использовании моей программой?
26. Как GPL сочетается со шрифтами?
27. Можно ли выпустить измененную версию программы под GPL только в двоичном виде?
28. Можно ли выпускать программу под лицензией, в которой сказано, что ее измененные версии распространять под GPL можно, а саму первоначальную версию — нельзя?
29. Является ли размножение и применение многих копий внутри одной организации или предприятия “распространением”?
30. Является ли распространением передача копии в контролируемое дочернее предприятие?
31. Можно ли в программе установки просить людей нажать кнопку в знак согласия с GPL?
32. Может кто-нибудь, кто передает программы под GPLv3 в пользовательском продукте, применять удаленную аттестацию для предотвращения модификации этих программ пользователем?
33. Что в GPLv3 означает выражение “правила и протоколы связи по сети”?

34. В каких случаях GPL распространяется и на результат работы программы под GPL?
35. Совместима ли GPLv3 с GPLv2?
36. Насколько различные лицензии GNU совместимы друг с другом?
37. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
38. Когда программа и ее внешние модули считаются единой комбинированной программой?
39. Можно ли применять GPL, когда пишется внешний модуль к несвободной программе?
40. Какие юридические проблемы возникают, если применяются несовместимые с GPL библиотеки с программами под GPL?

5.2.3. Оценочные средства для промежуточной аттестации

5.2.3.1 Примерные вопросы к экзамену

1. Как расшифровывается “GPL”?
2. Подразумевает ли выражение “свободная программа” применение GPL?
3. Почему мне следует применять GNU GPL, а не другие лицензии свободных программ?
4. Все ли программы GNU используют в качестве лицензии GNU GPL?
5. Если для программы применяется GPL, то делает ли это ее программой GNU?
6. Можно ли применять GPL для чего-нибудь кроме программ?
7. Почему некоторые библиотеки GNU выпускаются по обычной GPL, а не по Меньшей GPL?
8. Кто вправе требовать соблюдения GPL?
9. Почему ФСПО требует, чтобы те, кто вносит вклад в программы, авторские права на которые принадлежат ФСПО, передавали фонду авторские права?
10. Можно изменить GPL и сделать другую лицензию?
11. Почему GPL разрешает пользователям публиковать свои измененные версии?
12. Требуется ли GPL, чтобы исходный текст измененных версий выставлялся на всеобщее обозрение?
13. Может ли быть программа под GPL и не связанная с ней несвободная программа на одном и том же компьютере?
14. Что означает в GPL версии 2 “письменное предложение, действительное для любой третьей стороны”? Значит ли это, что все на свете могут получить исходный текст любой программы под GPL, все равно, какой именно?
15. В версии 2 GPL сказано, что если выпускаются измененные версии, то они должны быть “лицензированы... для всех третьих сторон”. Кто эти третьи стороны?
16. Позволяет ли GPL продавать копии программы за деньги?

17. Позволяет ли GPL распространять копии по договору о неразглашении?
18. Почему GPL требует передачи копии GPL в каждой копии программы?
19. Требуется ли заявление авторских прав на изменения в программе под GPL?
20. Что GPL говорит о переводе программы на другой язык программирования?
21. Если программа сочетает тексты из общественного достояния с текстами под GPL, можно ли взять часть из общественного достояния и пользоваться ею, как текстом из общественного достояния?
22. Позволяет ли GPL добавить требование ссылаться или размещать благодарности в научных статьях, где используются программы под GPL или результаты их работы?
23. Что имеют в виду, когда говорят, что две лицензии “совместимы”?
24. Что означают слова “лицензия совместима с GPL”?
25. Почему первоначальная лицензия BSD несовместима с GPL?
26. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
27. При определении того, составляют ли две части программы единое произведение, имеет ли какое-то значение факт, что программы заключаются в один или более контейнеров?
28. Почему ФСПО требует, чтобы те, кто вносит вклад в программы, авторские права на которые принадлежат ФСПО, передавали фонду авторские права?
29. Можно ли воспользоваться GPL как лицензией для аппаратуры?
30. Считается ли изменением предварительная компоновка исполняемого файла под GPL с различными библиотеками в системе для оптимизации его характеристик?
31. Как LGPL работает с Java?
32. Представляет ли собой “передача” в GPLv3 то же, что в GPLv2 подразумевается под “распространением”?
33. В GPLv3 “предоставление в общий доступ” приводится как пример перемещения. Что это значит? Является ли такое предоставление формой передачи?
34. Поскольку в рамках GPLv3 распространение и предоставление в общий доступ — это формы перемещения, которые являются также передачей, не могли бы вы привести примеры перемещения, которое передачей не является?
35. Что такое тивоизация? Как GPLv3 ее предотвращает?
36. Запрещает ли GPLv3 цифровое управление ограничениями?
37. Требуется ли GPLv3, чтобы у избирателей была возможность изменять программы, выполняющиеся на устройствах для голосования?
38. Есть ли в GPLv3 “параграф о защите от патентных исков”?

39. Что имеется в виду, когда в GPLv3 и AGPLv3 говорится “несмотря на все другие условия этой лицензии”?
40. Что в AGPLv3 считается “взаимодействующим с [программой] удаленно по компьютерной сети”?
41. Как понятие, выражаемое в GPLv3 словом “вы”, соотносится с определением “юридического или другого лица” в версии 2.0 лицензии Apache?
42. Что означает слово “данная программа” в GPLv3? Означает ли оно всякую программу, когда-либо выпущенную под GPLv3?
43. Если некоторая клиентская сетевая программа выпускается под AGPLv3, должна ли она быть в состоянии предоставить исходный текст серверам, с которыми она взаимодействует?
44. Как обновить (L)GPLv2 до (L)GPLv3?
45. Почему следует применять GNU GPL, а не другие лицензии свободных программ?
46. Почему GPL требует передачи копии GPL в каждой копии программы?
47. Достаточно ли просто добавить копию GNU GPL в репозиторий, чтобы применить ее к программам?
48. Почему нужно помещать уведомление о лицензии в каждый файл исходного текста?
49. Может ли разработчик программы, распространявший ее по GPL, впоследствии передать исключительные права на нее другой стороне?
50. Почему в программах должно быть сказано “версия 3 GPL или любая более поздняя версия”?
51. Что вы думаете о применении формулировки, в которой говорится, что определенная программа может быть использована только по самой поздней версии GNU GPL?
52. Есть ли способ распространить GPL на выходные данные, которые люди получают при использовании моей программой?
53. Как GPL сочетается со шрифтами?
54. Можно ли выпустить измененную версию программы под GPL только в двоичном виде?
55. Можно ли выпускать программу под лицензией, в которой сказано, что ее измененные версии распространять под GPL можно, а саму первоначальную версию — нельзя?
56. Фирма исполняет на сайте модифицированную версию программы под GPL. Сказано ли в GPL, что они должны выпустить свои модифицированные исходные тексты?
57. Является ли размножение и применение многих копий внутри одной организации или предприятия “распространением”?
58. Если кто-нибудь украдет компакт-диск с версией программы под GPL, дает ли GPL ему право распространять эту версию дальше?

59. Что, если предприятие передает копию своей собственной работы, распространяемой под GPL, как коммерческую тайну?
60. Является ли распространением передача копии в контролируемое дочернее предприятие?
61. Можно ли в программе установки просить людей нажать кнопку в знак согласия с GPL?
62. Если пользователь решит взять программы проекта на условиях GPLv3, значит ли это, что этому пользователю автоматически предоставлена содержащаяся в GPLv3 явная патентная лицензия?
63. Может кто-нибудь, кто передает программы под GPLv3 в пользовательском продукте, применять удаленную аттестацию для предотвращения модификации этих программ пользователем?
64. Что в GPLv3 означает выражение “правила и протоколы связи по сети”?
65. Поставщики, предоставляющие согласно GPLv3 сведения по установке, не обязаны предоставлять “услуги по поддержке” этого продукта. Какого рода “услуги по поддержке” вы имеете в виду?
66. Есть ли способ распространить GPL на выходные данные, которые люди получают при использовании моей программой? Например, если программа применяется для разработки аппаратуры, можно ли потребовать, чтобы эти разработки были свободны?
67. В каких случаях GPL распространяется и на результат работы программы под GPL?
68. Совместима ли GPLv3 с GPLv2?
69. Есть ли в GPLv2 требование о предоставлении информации об установке?
70. Насколько различные лицензии GNU совместимы друг с другом?
71. Какова разница между “размещением на одном носителе” и другими видами “измененных версий”?
72. Отличаются ли требования GPL к статической компоновке модулей с производением, на которое распространяется GPL, от требований к динамической компоновке?
73. Если библиотека выпускается на условиях GPL (не LGPL), значит ли это, что любая программа, которая ею пользуется, должна выпускаться на условиях GPL или совместимой с GPL лицензии?
74. Если интерпретатор языка программирования выпускается под GPL, значит ли это, что программы для него должны выпускаться на условиях совместимых с GPL лицензий?
75. Если лицензия интерпретатора языка программирования несовместима с GPL, можно ли выполнять на нем программы под GPL?
76. Когда программа и ее внешние модули считаются единой комбинированной программой?

77. Можно ли применять GPL, когда пишется внешний модуль к несвободной программе?
78. Можно ли выпустить несвободную программу, составленную так, чтобы подгружать внешний модуль под GPL?
79. Можно ли написать свободную программу, которая пользуется несвободными библиотеками?
80. Можно ли компоновать программу под GPL с несвободной системной библиотекой?
81. Какие юридические проблемы возникают, если применяются несовместимые с GPL библиотеки с программами под GPL?
82. Если в лицензии для модуля М есть требование, несовместимое с GPL, но требование действует, только когда М распространяется сам по себе, а не когда он включается в более крупную программу, делает ли это лицензию несовместимой с GPL? Можно ли комбинировать или связывать М с программой под GPL?
83. Если в таком объектно-ориентированном языке, как Java, я использую класс под GPL без изменений и образу из него подкласс, каким образом GPL влияет на более крупную программу?
84. Является ли нарушением GPL распространение несвободного драйвера, предназначенного для компоновки с ядром Linux?
85. Как позволить компоновать несвободные модули с библиотекой под GPL только по определенному протоколу?
86. Можно ли пользоваться отрывками исходных текстов под GPL в документации, которая лицензирована несовместимым с GPL образом?
87. Что следует делать, если обнаруживается возможное нарушение GPL? Кто вправе требовать соблюдения GPL?
88. Связан ли разработчик программы под GPL текстом лицензии? Могут ли быть действия разработчика нарушением GPL?
89. Можно ли использовать программы под GPL в устройстве, которое перестанет работать, если клиенты не будут вносить периодические платежи?
90. Что значит “устранить” нарушение GPLv3?
91. Если кто-то установит программу под GPL на портативном компьютере, а затем одолжит этот компьютер знакомому без предоставления исходного текста программы, будет ли это нарушением GPL?

5.3. Шкалы оценки образовательных достижений

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Шкала каждого контрольного мероприятия лежит в пределах от 0 до установленного максимального балла включительно. Итоговая аттестация по дисциплине оценивается по 100-балльной шкале и представляет собой сумму баллов, заработанных студентом при выполнении заданий в рамках текущего и промежуточного контроля.

Итоговая оценка выставляется в соответствии со следующей шкалой:

Сумма баллов	Оценка по 4-ех балльной шкале	Оценка ECTS	Требования к уровню освоению учебной дисциплины
90-100	5 – «отлично»	A	Оценка «отлично» выставляется студенту, если он глубоко и прочно усвоил программный материал, исчерпывающе, последовательно, четко и логически стройно его излагает, умеет тесно увязывать теорию с практикой, использует в ответе материал монографической литературы.
85-89	4 – «хорошо»	B	Оценка «хорошо» выставляется студенту, если он твёрдо знает материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос.
75-84		C	
70-74		D	
65-69	3 «удовлетворительно»	E	Оценка «удовлетворительно» выставляется студенту, если он имеет знания только основного материала, но не усвоил его деталей, допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала.
60-64			
Ниже 60	2 «неудовлетворительно»	F	Оценка «неудовлетворительно» выставляется студенту, который не знает значительной части программного материала, допускает существенные ошибки. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение без дополнительных занятий по соответствующей дисциплине.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

ОСНОВНАЯ ЛИТЕРАТУРА

1. GNU GENERAL PUBLIC LICENSE. Version 1, 1989 Free Software Foundation, Inc.
2. GNU GENERAL PUBLIC LICENSE. Version 2, 1991 Free Software Foundation, Inc.
3. GNU GENERAL PUBLIC LICENSE. Version 3, 2007 Free Software Foundation, Inc.
4. Andrew M. St. Laurent, Understanding Open Source and Free Software Licensing

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. GNU GENERAL PUBLIC LICENSE. Version 1, 1989 Free Software Foundation, Inc.
2. GNU GENERAL PUBLIC LICENSE. Version 2, 1991 Free Software Foundation, Inc.
3. GNU GENERAL PUBLIC LICENSE. Version 3, 2007 Free Software Foundation, Inc.
4. Andrew M. St. Laurent, Understanding Open Source and Free Software Licensing

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Операционная система GNU и движение за свободные программы <https://www.gnu.org/>
2. Документация проекта GNU <https://www.gnu.org/doc/other-free-books.html>
3. GNU LINUX — подробный обзор операционной системы <https://linuxvsem.ru/about-linux/gnu-linux-obzor>
4. Каталог свободных программ https://directory.fsf.org/wiki/Main_Page

LMS И ИНТЕРНЕТ-РЕСУРСЫ:

1. Обзор открытых лицензий https://www.opennet.ru/base/sys/open_licenses.txt.html
2. <https://www.jetbrains.com/ru-ru/pycharm/>
3. <https://docs.python.org/3/>
4. <https://stackoverflow.com/>

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Материально-техническое обеспечение включает в себя специально оборудованные аудитории и кабинеты. Компьютерные классы, аудитории оснащены мультимедийными средствами обучения.

Практические занятия проводятся в компьютерных классах, компьютеры подключены к сети Интернет.

Минимально необходимые требования:

- ✓ ОС Linux (Arch/Deb)
- ✓ Python 3.10
- ✓ JetBrains PyCharm Community

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

В соответствии с требованиями ОС ВО по направлению подготовки 09.03.02 «Информационные системы и технологии» реализация компетентного подхода предусматривает широкое использование в учебном процессе активных и интерактивных форм проведения занятий в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков студентов.

В рамках учебного курса студенты работают с лекциями, рекомендованной литературой, выполняют лабораторные работы, готовятся к экзамену. В процессе подготовки студенты используют программные продукты, инструментальные среды, информационно-справочные системы, информационные источники, размещенные в сети Интернет (официальные сайты, веб-порталы, тематические форумы и телекоммуникации), электронные учебники и учебно-методические пособия.

Для достижения планируемых результатов освоения дисциплины «Разработка программных средств на основе GNU GPL» используются следующие образовательные технологии:

- ✓ использование мультимедийного оборудования при проведении занятий;
- ✓ получение студентом необходимой учебной информации под руководством преподавателя и самостоятельно;
- ✓ проблемные лекции;
- ✓ «работа в команде» - совместная деятельность под руководством лидера, направленная на решение общей поставленной задачи;
- ✓ «междисциплинарное обучение» - использование знаний из разных областей, группируемых и концентрируемых в контексте конкретно решаемой задачи;
- ✓ контекстное обучение;
- ✓ обучение на основе опыта;
- ✓ разбор конкретных постановок экспериментов с поэтапным анализом процесса и обсуждением конечного результата;
- ✓ консультации;
- ✓ «индивидуальное обучение» - выстраивание для студента собственной образовательной траектории с учетом интереса и предпочтения студента;

- ✓ опережающая самостоятельная работа - изучение студентами нового материала до его изложения преподавателем на лекции и других аудиторных занятиях;
- ✓ подготовка к олимпиадам и к докладам на студенческих конференциях.

По дисциплине «Разработка программных средств на основе GNU GPL» в рабочем учебном плане предусмотрены интерактивные часы для проведения лабораторных работ. Для реализации интерактивных форм обучения используются учебно-методические материалы, разработанные сотрудниками кафедры «Цифровых технологий».

9. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОРГАНИЗАЦИИ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Учебным планом на изучение дисциплины отводится один семестр. В конце семестра предусмотрен экзамен.

При преподавании дисциплины методически целесообразно в каждом разделе курса выделить наиболее важные моменты и акцентировать на них внимание обучаемых. Такие моменты отражены в изложенных выше пунктах, касающихся формируемых знаний студентов и их проверки.

При проведении практических занятий студентам прививаются также навыки работы с научной и учебно-методической литературой.

Обязательным является самостоятельная работа студентов дома и в аудитории под руководством преподавателя, выполнение индивидуальных заданий, посещение международных и всероссийских конференций.

Каждая лекция курса является изложением отдельной темы. Поэтому целесообразно проводить проработку материала после каждой лекции. Это позволит сразу усваивать материал и не откладывать возникающие вопросы, а решать непосредственно после занятия.

Эффективным методом обучения является проведение лабораторных работ и проверка их преподавателем.

Необходимо отметить особенности лекционного материала данного курса, указать, с основами каких предметов должен быть знаком студент к моменту изучения данной дисциплины, какими основными понятиями, методами и представлениями должен владеть студент, начиная изучение данной дисциплины.

Так как учебным планом предусмотрены лабораторные занятия, целесообразно акцентировать внимание студентов на необходимости дальнейшего использования полученных знаний при изучении последующих курсов, выполнении курсовых работ и защиты бакалаврской диссертации.

Программа составлена в соответствии с требованиями ОС ВО НИЯУ МИФИ к обязательному минимуму содержания основной образовательной программы по направлению подготовки 09.03.02 «Информационные системы и технологии»

Программу составил:

Рецензент: заведующий кафедрой ЦТ

О.В. Кривошеев