|  |
| --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  федеральное государственное АВТОНОМНОЕ образовательное учреждение высшего образования  «Национальный исследовательский ядерный университет «МИФИ» |
| **Обнинский институт атомной энергетики –**  филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»  **(ИАТЭ НИЯУ МИФИ)** |

|  |
| --- |
| **УТВЕРЖДАЮ** |
| Начальник отделения интеллектуальных кибернетических систем  \_\_\_\_\_\_\_\_\_\_\_\_ С.О.Старков |
| «\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г. |

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

|  |
| --- |
| Технологии программирования |
| *название дисциплины* |
|  |
| для студентов направления подготовки |
|  |
| 09.03.02 Информационные системы и технологии |
|  |
|  |
|  |
| программа |
| Информационные технологии |
|  |
|  |
|  |
| Форма обучения: очная |

**г. Обнинск 2021 г.**

Программа составлена в соответствии с образовательным стандартом высшего образования НИЯУ МИФИ по направлению подготовки 09.03.02 Информационные системы и технологии

Программу составил:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.А.Мирзеабасов, к.т.н., доцент ОИКС

Рецензент:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Программа рассмотрена на заседании отделения интеллектуальных кибернетических систем (О)

(протокол № \_\_\_\_ от «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2021 г.)

Руководитель образовательной программы

09.03.02 Информационные системы и технологии

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.А.Мирзеабасов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2021 г.

|  |  |
| --- | --- |
|  |  |

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

|  |  |  |
| --- | --- | --- |
| Коды компетенций | **Результаты освоения ООП**  **Содержание компетенций** | **Перечень планируемых результатов обучения по дисциплине** |
| ОПК-2 | Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности | Знать: виды программных средств, необходимых для разработки ПО  Уметь: программно реализовывать алгоритмы и структуры данных  Владеть: навыками работы в интегрированных средах разработки, способами отладки и тестирования ПО |
| ПК-6 | Способен разрабатывать технические спецификации на программные компоненты и осуществлять их реализацию | Знать: виды проектной документации, структуру технического задания на разработку ПО, методы проектирования ПО, способы представления проектных концепций с помощью языка UML  Уметь: анализировать постановку задачи, составлять ТЗ на разработку ПО, применять методы проектирования при разработке программных средств  Владеть: навыками подготовки тех-нической документации в соответствии с требованиями |

2. Место дисциплины в структуре ОПОП бакалавриата

Дисциплина реализуется в рамках части, формируемой участниками образовательных отношений.

Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин: программирование, объектно-ориентированное программирование, основы проектной деятельности, управление данными.

Дисциплины и/или практики, для которых освоение данной дисциплины необходимо как предшествующее: учебная практика (научно-исследовательская работа), производственная практика (научно-исследовательская работа).

Дисциплина изучается на 3 курсе в 6 семестре.

3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

|  |  |  |
| --- | --- | --- |
| **Вид работы** | **Форма обучения** (вносятся данные по реализуемым формам) | |
| **Очная** | |
| **Семестр** | **Курс** |
| **№ 5** | **№ 3** |
| **Количество часов на вид работы:** | |
| **Контактная работа обучающихся с преподавателем** |  | |
| **Аудиторные занятия *(всего)*** | **64** | |
| В том числе: |  | |
| *лекции*  *(лекции в интерактивной форме)* | 16 | |
| *практические занятия*  *(практические занятия в интерактивной форме)* | 16 | |
| *лабораторные занятия* | 32 | |
| **Промежуточная аттестация** |  | |
| В том числе: |  | |
| *курсовая работа (зачет)* | **-** | |
| *экзамен* | 36 | |
| **Самостоятельная работа обучающихся** |  | |
| **Самостоятельная работа обучающихся *(всего)*** | 80 | |
| В том числе: |  | |
| *проработка учебного (теоретического) материала* | 16 | |
| *выполнение курсовой работы* | 32 | |
| *подготовка ко всем видам контрольных испытаний текущего контроля успеваемости* | 16 | |
| *подготовка ко всем видам контрольных испытаний промежуточной аттестации* | 16 | |
|  |  | |
| **Всего (часы):** | **180** | |
| **Всего (зачетные единицы):** | **5** | |

4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № п/п | Наименование раздела /темы дисциплины | Виды учебной работы в часах (вносятся данные по реализуемым формам) | | | | |
| Очная форма обучения | | | | |
| Лек | Пр | Лаб | Внеауд | СРО |
| 1. | Основные этапы и принципы создания программных продуктов | 5 | 5 | 10 | - | 30 |
| 1.1 | Введение в технологию программирования | 1 | 1 | - | - | 10 |
| 1.2 | Обзор языка программирования Java | 2 | 2 | 8 | - | 10 |
| 1.3 | Проектирование программ | 2 | 2 | 6 | - | 10 |
| 2. | Базовые концепции технологии програм-мирования | 6 | 6 | 12 | - | 30 |
| 2.1 | Модели жизненного цикла ПО | 1 | 1 | - | - | 5 |
| 2.2 | Язык UML | 1 | 1 | - | - | 5 |
| 2.3 | **Системы поддержки версий** | 1 | 2 | 4 | - | 10 |
| 2.4 | **Дополнительные возможности Java** | 3 | 2 | 6 | - | 10 |
| 3. | Процессы жизненного цикла | 5 | 5 | 10 | - | 20 |
| 3.1 | Тестирование и отладка | 1 | 1 | 6 | - | 10 |
| 3.2 | Паттерны проектирования | 2 | 2 | 4 | - | 5 |
| 3.3 | Многопоточные приложения | 2 | 2 | - | - | 5 |
|  | **Всего:** | **16** | **16** | **32** | **-** | **80** |

*Прим.: Лек – лекции, Пр – практические занятия / семинары, Лаб – лабораторные занятия, Внеауд – внеаудиторная работа, СРО – самостоятельная работа обучающихся*

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Содержание | |
| 1. | Основные этапы и принципы создания программных продуктов | | |
| 1.1 | Введение в технологию программирования | Понятие технологии программирования. История. Структурный подход, рекомендации. Документирование программ. ЕСПД. Техническое задание: структура, содержание разделов. | |
| 1.2 | Обзор языка программирования Java | Язык Java. Типы данных: базовые, ссылочные. Операторы. Массивы. Классы. Наследование. Абстрактные классы и интерфейсы. Обработка исключений. Потоки ввода-вывода. Библиотека коллекций. | |
| 1.3 | Проектирование программ | Проектирование пользовательского интерфейса. Компоненты SWING. | |
| 2. | Базовые концепции технологии программирования | | |
| 2.1 | Модели жизненного цикла ПО | | Жизненный цикл ПО. Процессы ЖЦ. Классические модели ЖЦ. Экстремальное программирование. |
| 2.2 | Язык UML | Графическое представление концепций ООП. Виды диаграмм. Диаграммы классов, пакетов, компонентов. Варианты использования. Диаграммы развертывания. | |
| 2.3 | **Системы поддержки версий** | Версии исходного кода. Поддержка версий. Репозиторий. Импорт, обновление, фиксация изменений. Клиент-серверные и распределенные системы поддержки версий. Системы Subversion, Git. | |
| 2.4 | **Дополнительные возможности Java** | Графические возможности Java. Java2D.  Работа с базами данных. JDBC-драйверы. Connection, Statement, ResultSet. | |
| 3. | Процессы жизненного цикла | | |
| 3.1 | Тестирование и отладка | Принципы тестирования. Виды и стратегии тестирования. Понятие отладки. Классификация причин ошибок. Методы отладки. | |
| 3.2 | Паттерны проектирования | Обзор паттернов проектирования. Singleton, Facade, Adapter, Proxy. Model — View — Controller и компоненты JList, JTable | |
| 3.3 | Многопоточные приложения | Класс Thread. Интерфейсы Runnable, Callable, Future, ExecutorService. | |

Практические/семинарские занятия

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Содержание | |
| 1. | Основные этапы и принципы создания программных продуктов | | |
| 1.1 | Введение в технологию программирования | Обсуждение структуры проектов. Примеры задач. | |
| 1.2 | Обзор языка программирования Java | Примеры применения средств языка Java | |
| 1.3 | Проектирование программ | Разработка и обсуждение технического задания | |
| 2. | Базовые концепции технологии програм-мирования | | |
| 2.1 | Модели жизненного цикла ПО | | Обсуждение организации работы над коллективным проектом. |
| 2.2 | Язык UML | Построение UML-диаграмм | |
| 2.3 | **Системы поддержки версий** | Особенности организации коллективной работы с репозиторием исходного кода проекта | |
| 2.4 | **Дополнительные возможности Java** | Обсуждение организации работы с БД из Java-приложения | |
| 3. | Процессы жизненного цикла | | |
| 3.1 | Тестирование и отладка | Виды тестирования. Типичные ошибки проектирования и кодирования. | |
| 3.2 | Паттерны проектирования | Обсуждение паттернов проектирования | |
| 3.3 | Многопоточные приложения |  | |

Лабораторные занятия

|  |  |  |
| --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Название лабораторной работы |
| 1. | Основные этапы и принципы создания программных продуктов | |
| 1.2 | Обзор языка программирования Java | Лабораторная работа № 1: использование коллекций и потоков ввода вывода |
| 1.3 | Проектирование программ | Лабораторная работа № 1: проектирование пользовательского интерфейса |
| 2. | Базовые концепции технологии программирования | |
| 2.3 | **Системы поддержки версий** | Лабораторная работа №2: работа с репозиторием Git |
| 2.4 | **Дополнительные возможности Java** | Лабораторная работа №2: подключение к БД |
| 3. | Процессы жизненного цикла | |
| 3.1 | Тестирование и отладка | Лабораторная работа №3: тестирование и отладка |
| 3.2 | Паттерны проектирования | Лабораторная работа №4: шаблон проектирования MVC |

Курсовая работа

|  |  |  |
| --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Раздел курсовой работы |
| 1. | Основные этапы и принципы создания программных продуктов | |
| 1.1 | Введение в технологию программирования | Анализ предметной области. Постановка задачи. Разработка ТЗ. |
| 1.2 | Обзор языка програм-мирования Java | Выбор необходимых для решения задачи классов и пакетов Java |
| 1.3 | Проектирование программ | Проектирование пользовательского интерфейса с использованием компонентов SWING. |
| 2. | Базовые концепции технологии программирования | |
| 2.1 | Модели жизненного цикла ПО | Жизненный цикл ПО. Процессы ЖЦ. |
| 2.2 | Язык UML | Разработка необходимых для понимания проекта диаграмм: классов, пакетов, компонентов, вариантов использования. |
| 2.3 | Системы поддержки версий | Сохранение исходного кода проекта в репозитории. Организация коллективной разработки. |
| 2.4 | **Дополнительные возможности Java** | Использование графических возможности Java, доступа к базам данных. |
| 3. | Процессы жизненного цикла | |
| 3.1 | **Тестирование и отладка** | Анализ ошибок, выявленных при тестировании проекта. |
| 3.2 | **Паттерны проектирования** | Анализ использованных в рамках курсовой работы шаблонов проектирования |

Примерные темы курсовых работ:

- разработка монитора системных ресурсов

- разработка персонального органайзера

- разработка системы тестирования

- разработка графического редактора

- разработка файлового менеджера

- разработка пользовательского интерфейса для взаимодействия с СУБД

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

В качестве учебно-методических материалов используется рекомендованная литература (разделы 7 и 8), а также презентации по материалам лекций (в электронном виде).

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

6.1. Паспорт фонда оценочных средств по дисциплине

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Контролируемые разделы (темы) дисциплины (результаты по разделам)** | **Код контролируемой компетенции (или её части) / и ее формулировка** | **Наименование оценочного средства** |
| **Текущий контроль** | | | |
| 1.1 | Введение в технологию программирования | ПК-6 (уметь) | Проверка технического задания |
| 1.2 | Обзор языка програм-мирования Java | ОПК-2 (знать) | Контрольная работа №1 |
| ОПК-2 (уметь) | Лабораторная работа №1 |
| 1.3 | Проектирование программ | ОПК-2, ПК-6 (знать, владеть) | Контрольная работа №1 |
| 2.1 | Модели жизненного цикла ПО | ОПК-2 (знать) | Собеседование на практическом занятии |
| 2.2 | Язык UML | ПК-6 (знать) | Собеседование на практическом занятии |
| 2.3 | Системы поддержки версий | ОПК-2 (владеть) | Лабораторная работа №2 |
| 2.4 | **Дополнительные возможности Java** | ОПК-2 (уметь) | Лабораторная работа №2 |
| 3.1 | **Тестирование и отладка** | ОПК-2 (уметь, владеть) | Лабораторная работа №3, контрольная работа №2 |
| 3.2 | **Паттерны проектирования** | ОПК-2 (знать, владеть) | Собеседование на практическом занятии, лабораторная работа №4 |
| 3.3 | **Многопоточные приложения** | ОПК-2 (знать) | Собеседование на практическом занятии |
| **Промежуточный контроль** | | | |
| 1. | Введение в технологию программирования  Обзор языка программирования Java  Проектирование программ  Язык UML  Системы поддержки версий  Дополнительные возможности Java | ОПК-2 (знать, уметь, владеть)  ПК-6 (знать,уметь, владеть ) | Защита курсовой работы |
| 2. | Введение в технологию программирования  Обзор языка программирования Java  Модели жизненного цикла ПО  Язык UML  Системы поддержки версий  Дополнительные возможности Java  Тестирование и отладка  Паттерны проектирования  Многопоточные приложения | ОПК-2 (знать)  ПК-6 (знать) | Экзамен |

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

*6.2.1. Зачет*

Зачет с оценкой выставляется по результатам выполнения курсовой работы. Критерий оценки –полнота и адекватность программного проекта. Оценка включает в себя оценки за техническое задание (20%), правильное использование репозитория исходного кода и организацию коллективной работы (20%), соблюдение сроков выполнения проекта (10%), качество оформления кода - стиль кодирования, комментарии (10%), оценка уровня сложности проекта и соответствия выполненной работы той функциональности, которая была заявлена в ТЗ (40%).

*6.2.2. Экзамен*

Экзамен проводится в виде ответов на три вопроса билета. **Первый вопрос** билета:

1. Техническое задание на разработку программного обеспечения

2. Жизненный цикл разработки ПО. Процессы жизненного цикла в стандарте

ISO/IEC 12207.

3. Жизненный цикл разработки ПО. Каскадная модель ЖЦ.

4. Жизненный цикл разработки ПО. Спиральная модель ЖЦ.

5. Жизненный цикл разработки ПО. Быстрая разработка (RAD).

6. Жизненный цикл разработки ПО. Экстремальное программирование.

7. Поддержка версий исходного кода ПО. Работа с Subversion.

8. Тестирование ПО: определение, основные принципы.

9. Тестирование ПО: определение, стратегии формирования тестовых наборов.

10. Отладка. Классификация ошибок. Трудности отладки.

11. Отладка. Причины ошибок. Методы и средства отладки.

12. Поддержка версий исходного кода ПО. Работа с Git.

13. Паттерны проектирования. Singleton, Facade, Visitor.

14. Паттерны проектирования. Adapter, Proxy, Builder.

**Второй вопрос билета:**

15. Базовые и ссылочные типы данных Java. Массивы. Структура класса.

16. Наследование классов. Абстрактные классы. Интерфейсы.

17. Интерфейсы коллекций Java. Интерфейс List и его реализации — ArrayList, LinkedList.

18. Интерфейс множества Set. Реализации — HashSet

19. Ассоциативные массивы (словари, отображения) Map. Реализации: HashMap.

20. Исключения в Java (Exceptions). Генерация и обработка исключений.

21. Библиотека ввода-вывода в Java. InputStream - OutputStream. Фильтрация

ввода-вывода.

22. Библиотека ввода-вывода в Java. Reader - Writer. Фильтрация ввода-вывода.

23. Пользовательский интерфейс Java. Обработка событий.

24. Шаблон проектирования MVC (на примере компонента JList).

25. Шаблон проектирования MVC (на примере компонента JTable).

26. Организация доступа к базам данных в Java: технология JDBC. JDBC-

драйверы. Соединение с БД.

27. Организация доступа к базам данных в Java: технология JDBC. Выполнение

SQL-запросов. Statement, ResultSet.

28. Графика в Java: Java2D.

29. Интерфейсы в Java 8: статические методы, методы по умолчанию.

30. Java 8: лямбда-выражения. Функциональные интерфейсы

31. Использование Stream API. Промежуточные и терминальные операции.

**Третий вопрос билета:**

32. UML-диаграмма классов

33. UML-диаграмма пакетов

34. UML-диаграмма компонентов

35. UML-диаграмма вариантов использования.

36. UML-диаграмма развертывания.

Критерий оценки – правильность и полнота ответа на вопросы адекватность приведенных примеров. Оценка выставляется по шкале от 0 до 100% в следующих долях: первый и второй вопросы — до 40% каждый, третий вопрос — до 20%. Экзамен считается сданным при оценке не ниже 60%.

*6.2.3. Контрольная работа №1*

Контрольная работа предназначена для выявления качества усвоения теоретических и знаний и практических умений по следующим темам дисциплины: «Обзор языка программирования Java», «Проектирование программ». Эти разделы являются основополагающими для всей последующей работы по выполнению проектирования в рамках курсовой работы.

Контрольная работа включает в себя комплексное задание на разработку приложения с графическим интерфейсом пользователя, которое студент должен выполнить в виде java-проекта в среде NetBeans. В вариантах задания варьируется вид интерфейса пользователя, способ хранения данных и задачи их обработки.

*6.2.4. Контрольная работа №2*

Цель контрольной работы состоит в проверке навыков тестирования программного кода и проведения отладки. В основном, контрольная работа направлена на проверку знаний по разделу «Тестирование и отладка». Перед студентом ставится задача — проверить правильность реализации двух методов в заданном классе. Для этого необходимо выполнить следующие шаги: создать тестирующие классы с использованием библиотеки JUnit; проанализировать структуру тестируемого класса и описания проверяемых методов; создать содержательный тест для каждого метода; с помощью отладчика проанализировать возникающие ошибки и исправить их; добиться правильного прохождения созданных тестов.

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

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.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины.

Текущий контроль осуществляется два раза в семестр: контрольная точка № 1 (КТ № 1) и контрольная точка № 2 (КТ № 2).

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид контроля** | **Этап рейтинговой системы Оценочное средство** | **Балл** | |
| Минимум | Максимум |
| **Текущий** | **Контрольная точка № 1** | 24 | **40** |
| Техническое задание | 12 | 20 |
| Использование репозитория исходного кода | 12 | 20 |
| **Контрольная точка № 2** | 12 | 20 |
| Своевременность прохождения этапов | 6 | 10 |
| Качество исходного кода | 6 | 10 |
| **Промежуточный** | **Зачет (защита курсовой работы)** | 24 | 40 |
| **ИТОГО по дисциплине** | | 60 | 100 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Вид контроля** | **Этап рейтинговой системы Оценочное средство** | **Балл** | |
| Минимум | Максимум |
| **Текущий** | **Контрольная точка № 1** | 15 | **25** |
| Лабораторная работа №1 | 6 | 10 |
| Контрольная работа №1 | 9 | 15 |
| **Контрольная точка № 2** | 18 | 30 |
| Лабораторная работа №2 | 6 | 10 |
| Лабораторная работа №3 | 6 | 10 |
| Контрольная работа №2 | 9 | 15 |
| **Промежуточный** | **экзамен** | 24 | 40 |
| **ИТОГО по дисциплине** | | 60 | 100 |

Каждая их перечисленных выше процедур оценивания знаний, умений и навыков считается сданной при получении за нее балла, составляющего не менее 60% процентов от максимального за данное контрольное мероприятие.

Уровни  освоения знаний, умений и навыков (в % от максимального балла):

< 60% - неудовлетворительный;

60% - 69%  - удовлетворительный;

70% - 89% - хороший;

90% - 100% - отличный.

За несвоевременную сдачу любого из указанных в таблице оценочных средств оценка может быть снижена от 1 до 2 баллов.

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

а) основная учебная литература:

1. Липаев В.В. Проектирование и производство сложных заказных программных продуктов. М.:СИНТЕГ, 2011. - 400 с. (доступна на ЭБС Iqlib.ru)

2. Васильев А.Н. Самоучитель Java с примерами и программами. 4-е издание. Изд-во "Наука и Техника", 2017. - 368 c. URL: https://e.lanbook.com/book/101548

3. Oracle Java Tutorial. Trail: Learning the Java Language. 2021.

URL: https://docs.oracle.com/javase/tutorial/

4. Oracle Java Tutorial. Trail: Essential classes. 2021.

URL: https://docs.oracle.com/javase/tutorial/essential/index.html

5. Oracle Java Tutorial. Trail: Collections. 2021.

URL: https://docs.oracle.com/javase/tutorial/collections/index.html

6. Oracle Java Tutorial. Trail: JDBC Database Access.. 2021.

URL: https://docs.oracle.com/javase/tutorial/jdbc/index.html

7. Oracle Java Tutorial. Trail: 2D Graphics. 2021.

URL: https://docs.oracle.com/javase/tutorial/2d/index.html

б) дополнительная учебная литература:

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

Сайт фирмы Oracle, посвященный языку Java:

[www.oracle.com/technetwork/java/index.html](http://www.oracle.com/technetwork/java/index.html)

Сайт разработчиков среды NetBeans:

netbeans.org

Сайт официальной документации по языку Java:

docs.oracle.com/javase/tutorial/tutorialLearningPaths.html

Сайт Интернет-университета информационных технологий:

intuit.ru

9. Методические указания для обучающихся по освоению дисциплины

|  |  |
| --- | --- |
| **Вид учебного занятия** | **Организация деятельности студента** |
| Лекция | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на консультации, на практическом занятии и лабораторной работе. Уделить внимание следующим понятиям: структура ТЗ, основы языка программирования Java, процессы жизненного цикла, поддержка версий, организация тестирования ПО, отладка программ, диаграммы UML, паттерны проектирования. |
| Практические занятия | Проработка рабочей программы, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Работа с конспектом лекций, просмотр рекомендуемой литературы. Обсуждение требований к разрабатываемому проекту и содержанию ТЗ. Изучение классов и пакетов Java, необходимых для достижения целей выбранного проекта. |
| Курсовая работа | Общей задачей курсовой работы является выполнение коллективного проекта . Курсовая работа предполагает последовательную реализацию следующих этапов: постановка задачи, формирование рабочей группы, анализ требований, распределение обязанностей, составление технического задания, проектирование, программная реализация проекта, работа с репозиторием исходного кода, тестирование, сдача проекта.  Защита курсовой работы включает в себя проверку соответствия разработанного проекта поставленным задачам, оценку вклада каждого разработчика, оценку качества исходного кода. |
| Контрольная работа | Ознакомиться с основной и дополнительной литературой, включая справочные издания, зарубежные источники, основополагающие термины. Попрактиковаться в проектировании интерфейса, работе с потоками ввода-вывода, работе с системами поддержки версий, инструментальными средствами (средой разработки, отладчиком, системой тестирования). |
| Лабораторная работа | При выполнении лабораторных работ необходимо ориентироваться на конспекты лекций, примеры, обсуждаемые на практических занятиях, рекомендуемую литературу и др. |
| Подготовка к экзамену | При подготовке к экзамену необходимо ориентироваться на конспекты лекций, рекомендуемую литературу и др. |

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Средства разработки для языка Java — openjdk (распространяется бесплатно)

Среда разработки NetBeans (свободно распространяется, лицензии GPLv2 и CDDL)

Система поддержки версий Git (свободное ПО)

Библиотека тестирования JUnit (свободная, Eclipse Public License)

Свободные СУБД Firebird и PostgreSQL, развернутые на сервере

JDBC-драйверы для указанных СУБД (распространяются свободно)

Слайды для лекций готовятся в системе LaTeX (пакет Beamer).

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

Компьютерный класс сетевых технологий. Класс оснащен 10 компьютерами (Intel Core i5/8GB/1 TB) и 1 компьютером ( Intel Celeron 1.6 GHz, 2 GB RAM, 250 GB) с операционной системой OpenSUSE Linux, а также мультимедийным проектором. Есть доступ к WI-FI.

Аудиторный класс, оборудованный проекционным экраном, мультимедийным проектором и персональным компьютером (AMD, ATHLON64, 2.7 GHz, 4 GB RAM, 250 GB). Есть доступ к WI-FI.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Лекционные занятия проходят с обсуждением учебного материала, демонстрируемого в форме презентаций на экране с использованием мультимедиа-проектора. Значительная часть практических и лабораторных занятий также проводится в интерактивной форме при тесном контакте студентов с преподавателем.

В рамках лабораторных работ студенты получают практический опыт работы с инструментальными средствами разработки — средой разработки NetBeans, системой поддержки версий Subversion, библиотекой тестирования JUnit, отладчиком программного кода, а также проектируют и разрабатывают пользовательский интерфейс, реализуют доступ к БД из программного кода Java, знакомятся с возможностями графической библиотеки Java2D.

Проект, выполненный в рамках курсовой работы, представляется разработчиками на специальном занятии в виде доклада с презентацией и демонстрацией работающей системы.

12.2. **Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки**

|  |  |  |
| --- | --- | --- |
| № | Наименование раздела /темы дисциплины | Часть, изучаемая (осваиваемая) самостоятельно |
| 1. | Основные этапы и принципы создания программных продуктов | |
| 1.1 | Введение в технологию программирования | История технологии программирования. |
| 1.2 | Обзор языка програм-мирования Java | Расширенные возможности среды NetBeans |
| 1.3 | Проектирование программ | Компоненты JComboBox, JSpinner |
| 2. | Базовые концепции технологии програм-мирования | |
| 2.2 | Язык UML | Варианты использования. Диаграммы развертывания. |
| 2.4 | **Дополнительные возможности Java** | Использование JDBC-драйвера для выбранной СУБД. |
| 3. | Процессы жизненного цикла | |
| 3.2 | **Паттерны проектирования** | Шаблон MVC для компонентов JComboBox, JSpinner |

Контроль освоения самостоятельно изученного теоретического материала осуществляется в виде собеседования во время защиты лабораторных, в виде устного опроса на практических занятиях и экзамене, на открытом семинаре с презентациями проектов.

Кроме этого, студенты также самостоятельно выполняют большую часть предусмотренных практических работ, промежуточный результат которых представляется на практических и лабораторных занятиях, а конечный результат - на защите лабораторных и курсовой работ.

**Вопросы для самоконтроля:**

- для чего предназначен компонент JSpinner

- как используется шаблон MVC для компонента JComboBox

- как подключить JDBC-драйвер к разрабатываемому проекту

- какие сущности применяются при разработке диаграммы вариантов использования

12.3. Краткий терминологический словарь

**ТЗ** — техническое задание (на разработку программы или программного изделия)

**ПО** — программное обеспечение

**ЖЦ** — жизненный цикл ПО, время от появления замысла до завершения поддержки

**Репозиторий** — хранилище исходного кода, используемое системой поддержки версий

**Отладка** — процесс локализации и исправления ошибок в программе

**Тестирование** — запуск программы с целью обнаружения ошибок.