Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение   
высшего образования «Новосибирский национальный исследовательский  
государственный университет» (Новосибирский государственный университет, НГУ)

**Факультет информационных технологий**

СОГЛАСОВАНО

Декан ФИТ НГУ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.М. Лаврентьев

«03» июля 2019 г.

**Фонд оценочных средств промежуточной аттестации**

по дисциплине  **Объектно-ориентированное программирование на Java**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная Год обучения: 2, семестр 4

|  |  |
| --- | --- |
| Форма аттестации | Семестр |
| Экзамен | 4 |

Новосибирск 2019

**Фонд оценочных средств** промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Объектно-ориентированное программирование на Java», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Разработчики:

Старший преподаватель кафедры общей информатики ФИТ В.Ю. Рылов

Заведующий кафедрой общей информатики ФИТ,

доктор физико-математических наук Д.Е. Пальчунов

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,

кандидат технических наук А.А. Романенко

# Содержание и порядок проведения промежуточной аттестации по дисциплине

* 1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Объектно-ориентированное программирование на Java» проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

|  |  |  |  |
| --- | --- | --- | --- |
| Код | Компетенции, формируемые в рамках дисциплины | Семестр 4 | |
| Портфолио | Экзамен |
| ПКС-2 - Способность разрабатывать компоненты программных продуктов, в части следующих результатов обучения | | | |
| ПКС-2.1 | Владеть: навыками разработки программ на языках высокого уровня. | + | + |
| ПКС-2.7 | Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение | + | + |
| ПКС-2.9 | Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем | + | + |

Промежуточная аттестация включает 2 этапа. Все компетенций оценивается портфолио, в которое входят работы, выполненные в рамках дисциплины, и экзаменом.

Тематика экзаменационных вопросов и заданий экзамена включает следующие темы (разделы) в двух категориях:

Категория 1 – Основы объектного подхода

* Эволюция стилей и поколения языков программирования
* Основные принципы объектно-ориентированного подхода (абстракция, инкапсуляция, модульность, иерархия, типизация, параллелизм, сохраняемость)
* Объект с точки зрения ООП, взаимоотношения между объектами
* Класс с точки зрения ООП, взаимоотношения между классами
* Многоуровневая метамодель
* Принципы и шаблоны проектирования

Категория 2 – Средства объектно-ориентированного программирования Java

* Виртуальная машина Java
* Средства поддержки процедурного стиля в языке Java
* Средства поддержки объектно-ориентированного подхода в Java
* Средства поддержки обобщенного программирования в Java
* Стандартная библиотека Java
  1. Порядок проведения промежуточной аттестации по дисциплине

Промежуточная аттестация проводится в форме экзамена и включает 2 этапа: портфолио и экзамен. Для оценивания портфолио студенту необходимо сдать все работы, входящие в структуру портфолио.

Оценка отдельных элементов портфолио осуществляется по балловой системе.

При защите задания обучающийся должен:

* Изложить необходимый для решения теоретический материал
* Указать методику решения задания, предоставить диаграмму классов в нотации UML
* Предоставить исходный код программы или программ реализующий требования задания
* Продемонстрировать корректную работу программы и прохождения автоматических модульных тестов.

За решенные задания начисляются баллы в соответствии с условиями задач. В случае сдачи задания с недочетами или позже установленного срока начисляются баллы меньше балловой стоимости задачи но не менее 50% при условии успешного решения.

Коллоквиум состоит из 18-20 вопросов проводится в виде письменного теста, за каждый правильный ответ на вопрос коллоквиума начисляется от 1 до 2 баллов в зависимости от сложности вопроса.

Для получения оценки «зачтено» необходимо набрать не менее 65 баллов портфолио при типичной стоимости задачи от 10 до 30 баллов. Должно быть сдано не менее 4 заданий.

Необходимым условием для прохождения промежуточной аттестации является оценка «зачтено» по результатам выполненного портфолио.

Экзамен проводится в устной форме. Во время проведения экзамена студенту не разрешается использовать технические средства и литературу. В процессе ответа на вопросы экзаменационного билета студенту могут быть заданы дополнительные вопросы по темам дисциплины

# Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.2.

Таблица П1.2

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Наименование оценочного средства | Краткая характеристика оценочного средства | Представление оценочного средства в фонде |
| Семестр 4 | | | |
|  | Коллоквиум | Средство контроля усвоения учебного материала дисциплины, организованное как письменный тест | Вопросы по темам/разделам дисциплины |
|  | Портфолио | Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах. | Структура портфолио |
|  | Экзаменационный билет | Комплекс вопросов и/или разноуровневых заданий (списать подходящее из п. про задачи) | Список теоретических вопросов и задач |

2.1. Требования к структуре и содержанию оценочных средств  
аттестации

**2.1.1. Список вопросов коллоквиума**

1) Какие средства Java поддерживают принцип абстракции?

2) Какие средства Java обеспечивают поддержку инкапсуляции, расположите их в порядке открытости?

3) Определите понятие объекта. Перечислите варианты отношений между объектами, в чем состоит взаимосвязь поведения и состояния объекта?

4) Определите понятие класса. Перечислите варианты отношений между классами. Обозначьте взаимосвязь между этими отношениями, если она существует.

5) Сформулируйте принцип подстановки Лисковой.

6) В чем состоит поддержка принципа типизации в языке Java? Приведите пример кода безопасного приведения типов

7) Перечислите примитивные типы языка Java, которые вы знаете. Каков их диапазон допустимых значений?

8) Для чего нужен класс java.lang.Class. Как узнать класс объекта, если у вас есть ссылка типа Object o ?

9) Какие строки содержат ошибки и почему?

a. int[] a = new int[4];

b. int[] a = new int[];

c. int[4] a = new int[4];

d. int a[] = new int[4];

10) Какие строки содержат ошибки и почему?

a. final class A{ ...

b. abstract class A{ ...

c. protected class A{ ...

d. abstract int b;

e. protected int b(3);

11) Какой блок catch будет ловить все?

a. catch(Error e){}

b. catch(AbstractException e){}

c. catch(Throwable e){}

d. catch(Exception e){}

12) У вас есть следующая иерархия типов:

class A {

public void f() {...}

}

class B extends A {

public void f() {...}

}

class C extends B {

public void f() {

}

}

a. Вы имеете экземпляр C c = new C(); как вызвать метод B.f() из метода C.f()?

b. Как вызвать метод A.f() из метода C.f()?

13) Как правильно создать экземпляр класса Inner в методе main() класса Outer :

public class Outer {

public class Inner {}

public static void main() {

???

}

}

14) Что выведет на экран данная программа:

public class A{

public void method() throws Exception {

System.out.println("Шаг 2");

throw new Error("Ошибка");

}

public static void main(String[] args) {

A a = new A();

System.out.println("Шаг 1");

try {

a.method();

} catch (Exception ex) {

System.out.println("Исключение!");

}

System.out.println("Шаг 3");

}

}

15) Что выведет на экран данная программа:

public class Test {

public static void add3(Integer i) {

int val = i.intValue();

val += 3;

i = new Integer(val);

}

public static void main(String[] args) {

Integer i = new Integer (0);

add3(i);

System.out.printIn (i.intValue() );

}

}

16) Какое из следующих определений запрещает наследование класса FooBar:

a. static class FooBar { }

b. private class FooBar { }

c. abstract public class FooBar { }

d. final public class FooBar { }

e. final abstract class FooBar { }

17) Какое утверждение является верным:

a. Анонимный класс может быть объявлен статическим

b. Статический внутренний класс не может быть статическим членом внешнего класса

c. Статический внутренний класс не требует экземпляра объемлющего класса

d. На переменные экземпляра внутреннего статического класса можно ссылаться, используя имя этого класса

e. Для создания экземпляра статического внутреннего класса требуется экземпляр внешнего класса

18) Напишите цикл, который подсчитает среднее значение объектов-значений хранящихся в Map:

public class PrintAverage {

public static int average(Map<String, Integer> map) {

}

}

19) Напишите метод быстрого копирования из InputStream в OutputStream (не забудьте про исключения):

public class Copy {

public static void copy(InputStream srcStream, OutputStream destStream) {

}

**2.1.2 Требования к портфолио**

Портфолио должно содержать результаты участия в коллоквиуме (письменный тест) и 4-5 выполненных заданий по следующим темам:

* Алгоритмические средства Java, классы Object, String, ввод-вывод
* Шаблон проектирования «фабричный метод», журналирование, модульное тестирование
* Шаблон проектирования MVC. Графический интерфейс пользователя
* Многопоточность и параллелизм
* Распределенное программирование и сетевое взаимодействие

**2.1.3 Форма и перечень вопросов экзаменационного билета**

**Форма  экзаменационного билета**

Таблица П1.3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Новосибирский государственный университет  Экзамен | |  |
|  | Объектно-ориентированное программирование на Java | |  |
|  | наименование дисциплины  09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  Программная инженерия и компьютерные науки | |  |
|  | наименование образовательной программы    ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №  1. Вопрос из категории 1  2. Вопрос из категории 2 | |  |
|  | Составитель |  |  |
|  |  | В.Ю. Рылов |  |
|  | (подпись) |  |  |
|  | Ответственный за образовательную программу | |  |
|  |  | А.А. Романенко |  |
|  | (подпись) |  |  |
|  | «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20     г. |  |  |

Перечень вопросов экзамена, структурированный по категориям, представлен в таблице П1.4

|  |  |
| --- | --- |
| Категория, компетенции | Формулировка вопроса |
| **Категория 1 – Основы объектного подхода** | |
| ПКC -2, ПКC-2.1 | Эволюция методологий программирования. Парадигмы программирования |
| ПКC -2, ПКC-2.1 | Основные принципы объектного подхода. Абстрагирование |
| ПКC -2, ПКC-2.1 | Основные принципы объектного подхода. Инкапсуляция |
| ПКC -2, ПКC-2.1 | Основные принципы объектного подхода. Модульность |
| ПКC -2, ПКC-2.1 | Основные принципы объектного подхода. Иерархия |
| ПКC -2, ПКC-2.1 | Основные принципы объектного подхода. Типизация |
| ПКC -2, ПКC-2.1 | Объект с точки зрения ООП. Состояние. Поведение |
| ПКC -2, ПКC-2.1 | Объект с точки зрения ООП. Идентичность и жизненный цикл объектов |
| ПКC -2, ПКC-2.1 | Объект с точки зрения ООП. Взаимоотношения между объектами. |
| ПКC -2, ПКC-2.1 | Классы. Природа классов. Метамодель. Инстанцирование. |
| ПКC -2, ПКC-2.1 | Классы. Структура класса. Абстрактные классы и интерфейсы |
| ПКC -2, ПКC-2.9 | Классы. Принцип подстановки Лисковой. Принцип разделения интерфейсов |
| ПКC -2, ПКC-2.7 | Классы. Средства UML для построения диаграмм классов |
| ПКC -2, ПКC-2.1 | Классы. Отношения между классами. Ассоциация и агрегация |
| ПКC -2, ПКC-2.1 | Классы. Иерархии классов. Зависимость |
| ПКC -2, ПКC-2.9 | Принципы и шаблоны GRASP |
| ПКC -2, ПКC-2.9 | Шаблоны GOF |
| **Категория 2 – Средства объектно-ориентированного программирования на Java** | |
| ПКC -2, ПКC-2.1 | Виртуальная машина. Структура программ. Типы переменных в Java. Принципы работы ClassLoader |
| ПКC -2, ПКC-2.1 | Средства абстракции Java. Структура класса. Статические члены. Перечисления |
| ПКC -2, ПКC-2.1 | Внутренние и вложенные классы. Статический и динамический контекст. Локальные и анонимные классы |
| ПКC -2, ПКC-2.1 | Средства инкапсуляции Java. Поддержка модульности. Пакеты. |
| ПКC -2, ПКC-2.1 | Представление иерархических отношений. Наследование. Интерфейсы и абстрактные классы |
| ПКC -2, ПКC-2.1 | Агрегация и зависимость от времени жизни. Реализация отношений клиент-сервер. Стандартные контейнеры |
| ПКC -2, ПКC-2.1 | Типизация. Правила преобразования типов. instanceof и ClassCastException. Класс Class |
| ПКC -2, ПКC-2.1 | Родовые компоненты Java, отличия от шаблонов C++, ограничения на параметры-типы |
| ПКC -2, ПКC-2.1 | Исключения. Обработка исключительных ситуаций |
| ПКC -2, ПКC-2.1 | Система ввода-вывода, пакет java.io, пакет java.net |
| ПКC -2, ПКC-2.1 | Средства поддержки параллелизма. Активные и пассивные объекты. Класс Object |
| ПКC -2, ПКC-2.1 | Использование Thread и Runnable. Пул потоков, назначение и принципы реализации |
| ПКC -2, ПКC-2.1 | Сохраняемость. Serializable и Externalizable. Программирование распределенных приложений |
| ПКC -2, ПКC-2.1 | Модель безопасности Java. Policy, Permissions, AccessController |
| ПКC -2, ПКC-2.1 | Графическая подсистема. Основы AWT, Applet, Swing components. Событийная модель |

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Объектно-ориентированное программирование на Java» в текущем учебном году.

# Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине

Таблица П1.5

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Шифр компетенций | Структурные элементы оценочных средств | Показатель сформированности | Не сформирован («неудовлетворительно») | Пороговый уровень («удовлетворительно») | Базовый уровень («хорошо») | Продвинутый уровень («отлично») |
| ПКС-2 | Портфолио, экз. вопрос категории 1, экз. вопрос категории 2 | ПКС-2.1 Владеть: навыками разработки программ на языках высокого уровня | Знает отдельные объектно-ориентированные средства языка Java, не знает отличий виртуального и параметрического полиморфизма, модели наследования и преобразования типов | Знает основные средства ООП языка Java, знает функции управления жизненным циклом объекта, умеет реализовывать основные отношения между классами | Знает все средства ООП языка Java, виртуальный полиморфизм, виртуальное наследование, управление жизненным циклом объектов, умеет реализовывать все типы отношений между классами | Демонстрирует глубокие знания и особенности реализации на уровне языка ассемблера JVM |
| ПКС-2 | Портфолио, Экз. вопрос категории 1 | ПКС-2.7 Уметь: проводить объектную декомпозицию информационной системы, вырабатывать и обосновывать архитектурное решение | Не знает нотации, не умеет строить диаграммы UML для визуализации отношений между классами | Может представлять классы и их структуру, может представить диаграмму классов своей программы | Умеет декорировать (именовать, аннотировать, специфицировать мощность) отношения между классами, знает различия в обозначениях зависимости, ассоциации, агрегации и композиции | Уверенно строит диаграммы классов, решения практических задания портфолио снабжены диаграммами, демонстрирующими структуру программы |
| ПКС-2 | Портфолио, экз. вопрос категории 1 | ПКС-2.9 Знать: шаблоны проектирования и архитектурные шаблоны построения информационных систем | Не знает основных принципов и шаблонов проектирования | Знает основные принципы и шаблоны, не демонстрирует глубокого понимания, затрудняется применять их на практике | Знает основные принципы и шаблоны, может применять их на практике | Демонстрирует глубокие познания, знает особенности применения, активно применяет шаблоны и принципы при реализации заданий |

# Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в 4 семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно». Оценки «отлично», «хорошо», «удовлетворительно» означают успешное прохождение промежуточной аттестации.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

Оценка «неудовлетворительно» выставляется, если хотя бы одна компетенция не сформирована.

**Лист актуализации фонда оценочных средств промежуточной аттестации**

**по дисциплине  
«Объектно-ориентированное программирование на Java»**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Характеристика внесенных изменений (с указанием пунктов документа) | Дата и №  протокола Ученого совета ФИТ | Подпись  ответственного |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |