Министерство образования и науки Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования «Казанский национальный исследовательский технический университет им. А.Н. Туполева-КАИ»

Институт (факультет) **Институт компьютерных технологий и защиты информации**

Кафедра **Динамики процессов и управления**

УТВЕРЖДАЮ

Ответственный за ОП

_____ Р.Т. Сиразетдинов

« 31 » 08 20 17 г.

Регистрационный номер 6010-19(ф) ПЧ

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения промежуточной аттестации обучающихся по дисциплине (модулю) или практике

«Программирование на языках высокого уровня»

Индекс по учебному плану: **Б1.Б.14**

Направление подготовки: **09.03.03 «Прикладная информатика»**

Квалификация: **бакалавр**

Профиль подготовки: **Прикладная информатика в экономике**

Вид(ы) профессиональной деятельности: **научно-исследовательская, проектная**

Заведующий кафедрой ДПУ Р.Т. Сиразетдинов

Разработчик: доцент кафедры ДПУ Ш.Р.Сабитов

Казань 2017 г.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю) или практике

Программирование на языках высокого уровня

Содержание фонда оценочных средств (ФОС) соответствует требованиям федерального государственного стандарта высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 «Прикладная информатика», учебному плану направления 09.03.03 «Прикладная информатика».

Разработанные ФОС обладают необходимой полнотой и являются актуальными для оценки компетенций, осваиваемых обучающимися при изучении дисциплины «Программирование на языках высокого уровня». Разработанные ФОС полностью соответствуют задачам будущей профессиональной деятельности обучающихся, установленных ФГОС ВО по направлению 09.03.03 «Прикладная информатика». В составе ФОС присутствуют оценочные средства в виде тестовых заданий и контрольных вопросов различного уровня сложности, которые позволяют провести оценку порогового, продвинутого и превосходного уровней освоения компетенций по дисциплине.

ФОС обладают необходимой степенью приближенности к задачам будущей профессиональной деятельности обучающихся, связанным со способностью к самоорганизации и самообразованию, решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий в профессиональной деятельности.

Существенные недостатки отсутствуют.

Заключение. Учебно-методическая комиссия делает вывод о том, что представленные материалы соответствуют требованиям ФГОС ВО по направлению 09.03.03 «Прикладная информатика» и рекомендуются для использования в учебном процессе.

Рассмотрено на заседании учебно-методической комиссии «31» августа 2017 г., протокол № 8.

Председатель УМК _____Родионов В.В.

Содержание

ВВЕДЕНИЕ 4

1. ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ 5

2. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ 5

3. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ С УКАЗАНИЕМ ЭТАПОВ ИХ ФОРМИРОВАНИЯ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ 5

4. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ ОЦЕНИВАНИЯ КОМПЕТЕНЦИЙ НА РАЗЛИЧНЫХ ЭТАПАХ ИХ ФОРМИРОВАНИЯ, ОПИСАНИЯ ШКАЛЫ ОЦЕНИВАНИЯ 6

5. МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ, ОПРЕДЕЛЯЮЩИЕ ПРОЦЕДУРУ ОЦЕНИВАНИЯ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ 9

6 КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ, НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ, ХАРАКТЕРИЗУЮЩИХ ЭТАПЫ ФОРМИРОВАНИЯ КОМПЕТЕНЦИЙ В ПРОЦЕССЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ 11

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ И ДОПОЛНЕНИЙ 21

Введение

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Программирование на языках высокого уровня» – это комплект методических и контрольно-измерительных материалов, предназначенных для определения уровня сформированности компетенций, оценивания знаний, умений, владений на разных этапах освоения дисциплины для проведения промежуточной аттестации обучающихся по дисциплине.

ФОС ПА является составной частью учебного и методического обеспечения программы бакалавриата по специальности 09.03.03. «Прикладная информатика».

Задачи ФОС по дисциплине «Программирование на языках высокого уровня»:

– оценка запланированных результатов освоения дисциплины обучающимися в процессе изучения дисциплины, в соответствии с разработанными и принятыми критериями по каждому виду контроля;

– контроль и управление процессом приобретения обучающимися необходимых знаний, умений, навыков и формирования компетенций, определенных в ФГОС ВО по направлению подготовки

ФОС ПА по дисциплине дисциплине «Программирование на языках высокого уровня» сформирован на основе следующих основных принципов оценивания:

– пригодности (валидности) (объекты оценки соответствуют поставленным целям обучения);

– надежности (использования единообразных стандартов и критериев для оценивания запланированных результатов);

– эффективности (соответствия результатов деятельности поставленным задачам).

ФОС ПА по дисциплине «Программирование на языках высокого уровня» разработан в соответствии с требованиями ФГОС ВО по специальности 09.03.03. «Прикладная информатика» для аттестации обучающихся на соответствие их персональных достижений требованиям поэтапного формирования соответствующих составляющих компетенций и включает контрольные вопросы (или тесты) и типовые задания, необходимые для оценки знаний, умений и навыков, характеризующих этапы формирования компетенций.

1. Формы промежуточной аттестации по дисциплине

Дисциплина «Программирование на языках высокого уровня» изучается во 2 семестре при очной форме обучения и завершается промежуточной аттестацией в форме экзамена.

2. Оценочные средства для промежуточной аттестации

Оценочные средства для промежуточной аттестации по дисциплине «Программирование на языках высокого уровня» при очной форме обучения.

Таблица 1

Оценочные средств для промежуточной аттестации
(очная форма обучения)

№ п/п	Семестр	Форма промежуточной аттестации	Оценочные средства
1.	2	Экзамен	ФОС ПА

3. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Перечень компетенций и их составляющих, которые должны быть сформированы при изучении темы соответствующего раздела дисциплины «Программирование на языках высокого уровня», представлен в таблице 2.

Таблица 2

Перечень компетенций и этапы их формирования
в процессе освоения дисциплины

№ п/п	Этап формирования (семестр)	Наименование раздела	Код формируемой компетенции (составляющей компетенции)		Форма промежуточной аттестации
1	2	Объектно-ориентированное программирование. Принципы S.O.L.I.D.	ОПК-4 ОК-7	ОПК-4.З ОПК-4.У ОПК-4.В ОК-7.З	Экзамен
2	2	Рефакторинг. Паттерны. Лямбда.	ОПК-4	ОПК-4.З ОПК-4.У ОПК-4.В ОК-7.З ОК-7.У ОК-7.В	Экзамен

4. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описания шкалы оценивания

Показатели и критерии оценивания сформированности компетенций на зачете, приведены в таблице 3.

Таблица 3

Показатели и критерии оценивания сформированности компетенций на зачете

№ п/п	Этап формирования (семестр)	Код формируемой компетенции (составляющей компетенции)		Критерии оценивания	Показатели оценивания (планируемые результаты обучения)		
					Пороговый уровень	Продвинутый уровень	Превосходный уровень
1.	2	ОПК-4 ОК-7	ОПК-4.З ОПК-4.У ОК-7.З ОК-7.У	Теоретические навыки	- Неполное знание информационно-коммуникационные технологии, применяемые для решения стандартных задач профессиональной деятельности. - Неполное умение учитывать основные требования информационной безопасности при решении профессиональных задач	- В целом сформировавшееся знание информационно-коммуникационные технологии, применяемые для решения стандартных задач профессиональной деятельности. - В целом сформировавшееся умение учитывать основные требования информационной безопасности при решении профессиональных задач	- Сформировавшееся систематическое знание информационно-коммуникационные технологии, применяемые для решения стандартных задач профессиональной деятельности. - Сформировавшееся систематическое умение учитывать основные требования информационной безопасности при решении профессиональных задач

2.	2	ОПК-4	ОПК-4.З ОПК-4.У ОПК-4.В ОК-7.З ОК-7.У ОК-7.В	Практические навыки	- Неполное владение способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	- В целом сформировавшееся владение способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	- Сформировавшееся систематическое владение способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности

Формирование оценки при промежуточной аттестации по итогам освоения дисциплины зависит от уровня освоения компетенций, которые обучающийся должен освоить по данной дисциплине. Связь между итоговой оценкой и уровнем освоения компетенций (шкала оценивания) представлена в таблице 4.

Таблица 4

Описание шкалы оценивания

Шкала оценивания		Описание оценки в требованиях к уровню и объему компетенций
Словесное выражение	Выражение в баллах	
Зачтено	от 86 до 100	Освоен **превосходный** уровень всех компетенций (составляющих компетенций)
Зачтено	от 71 до 85	Освоен **продвинутый** уровень всех компетенций (составляющих компетенций)
Зачтено	от 51 до 70	Освоен **пороговый** уровень всех компетенций (составляющих компетенций)
Не зачтено	до 51	Не освоен **пороговый** уровень всех компетенций (составляющих компетенций)

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

Формирование оценки по результатам текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Программирование на языках высокого уровня» приведено в таблице 5.

Таблица 5

Формирование оценки по итогам освоения дисциплины

Наименование контрольного мероприятия	Рейтинговые показатели				
	I аттестация	II аттестация	III аттестация	по результатам текущего контроля	по итогам промежуточной аттестации (зачета /экзамена)
Раздел 1. Объектно-ориентированное программирование. Принципы S.O.L.I.D.	**20**			**20**	
Тест текущего контроля по разделу	10			10	
Защита лабораторных работ	10			10	
Раздел 2. Рефакторинг. Паттерны. Лямбда.		**30**		**30**	
Тест текущего контроля по разделу		10		10	
Защита лабораторных работ		20		20	
Промежуточная аттестация (Экзамен):					**50**
– тест промежуточной аттестации по дисциплине					20
– ответы на контрольные вопросы в письменной форме					30

6 Контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения дисциплины

6.1 Тестовые задания

Тест по текущему контролю раздела 1.

1. Структура отличается от класса тем, что все ее члены

a. не являются функциями
b. могут быть только базового типа
c. являются общими
d. являются защищенными

2. Объединенное в единое целое множество поименованных элементов разных типов есть

a. массив
b. структура
c. класс
d. файл
e. ссылка

3. Аргумент функции — это:

a. переменная функции, получающая значение из вызывающей программы;
b. способ, с помощью которого функция защищает себя от воздействия значений, передаваемых вызывающей программой;
c. значение, передаваемое вызывающей программой в функцию;
d. значение, возвращаемое функцией вызывающей программе.

4. Объектно-ориентированное программирование характеризуется:

a. Наличием одной линейной программы

b. Разделением программы на модули
c. все данные об объекте, его связи с другими объектами объеденяються в одну структурную переменную

5. Метод это:
a. Функция, получая в качестве обязательного параметра указатель на объект
b. Структура, хранящая указатели this, parrent, supper
c. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
d. Определенный программистом абстрактный тип данных

6. Объект это:
a. Функция, получая в качестве обязательного параметра указатель на объект
b. Структура, хранящая указатели this, parrent, supper
c. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
d. Определенный программистом абстрактный тип данных

7. Класс это:
a. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
b. Определенный программистом абстрактный тип данных
c. Переменная, описанная абстрактным типом данных

8. Когда данные являются не глобальными, доступными всей программе, а локальными доступными только малой части программы:
a. При инкапсуляции
b. При полиморфизме
c. При объявлении метода

9. Когда функция определяется независимо в каждом производном классе и имеет в этих классах общее имя:

a. При инкапсуляции
b. При полиморфизме
c. При наследовании и инкапсуляции

10. Для выполнения операции присвоение одного объекта другому:

a. Достаточно чтобы в класе этих обхектов был специальный конструктов — конструктор копирования
b. Классы этих объектов нужно описать одинакомым образом
c. Достаточно чтобы в класе этих объектов были конструкторы и деконструкторы
d. Необходимо описать оба объекта одним именем класса и иметь в классе конструктор копирования

Тест текущего контроля по разделу 2.

1. Базовые принципы построения объектов в ООП

a. Наследование
b. Инкапсуляция
c. Инсталляция
d. Пролонгация
e. Полиморфизм

2. Объектно-ориентированное программирование характеризуется

a. Наличием одной линейной программы
b. Разделением программы на модули
c. все данные об объекте, его связи с другими объектами объеденяются в одну структурную переменную

3. Метод это:

a. Функция, получая в качестве обязательного параметра указатель на объект
b. Структура, хранящая указатели this, parrent, supper
c. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
d. Определенный программистом абстрактный тип данных

4. Объект это:

a. Функция, получая в качестве обязательного параметра указатель на объект
b. Структура, хранящая указатели this, parrent, supper
c. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
d. Определенный программистом абстрактный тип данных

5. Класс это:

a. Структурная переменная, содержащая всю информацию о некотором физическом предмете или реализуемом в программе понятии
b. Определенный программистом абстрактный тип данных
c. Переменная, описанная абстрактным типом данных

6. Когда данные являются не глобальными, доступными всей программе, а локальными доступными только малой части программы:

a. При инкапсуляции

b. При полиморфизме

c. При объявлении метода

7. Для выполнения операции присвоение одного объекта другому:
a. Достаточно чтобы в классе этих объектов был специальный конструктов - конструктор копирования
b. Классы этих объектов нужно описать одинаковым образом
c. Достаточно чтобы в классе этих объектов были конструкторы и деконструкторы
d. Необходимо описать оба объекта одним именем класса и иметь в классе конструктор копирования

8. В каких случаях надо иметь в классе конструктор копирования:
a. Когда нужно передать элементы класса
b. Для выполнения операции присвоение одного элемента другому
c. Для создания дружественной функции
d. Для уничтожения объекта из памяти

9. В чем состоит задача конструкттора копирования:

a. Четко определить порядок создания копии объекта

b. Размножать объект в указанном колличестве копий

c. Копировать свойства других объектов

d. Создать this указатель

10. Являеться ли конструктор Array(int n=0) ошибкой

a. Да

b. Нет

11. Какое ключевое слово должно быть использовано чтобы указать что объект не является модифицируемым и любая попытка изменения этого типа является ошибкой:

a. readonly

b. const

c. static

d. undeclare

12. **Как называются элементы класса которые относятся ко всем** экземплярам объектов класса

a. Статические

b. Динамические

c. Константные

d. Защищенные

13. В каких случаях следует вводить статические элементы массива:

a. Когда требуется контроль колличества общих методов класса

b. Когда требуется доступ к элементам массива из объектов дружественного класса

c. Когда требуется одновременный доступ ко всем объектам класса

d. Когда требуется определить изменяемые элементы класса

14. При обращении к нестатическому методу перед его именем указывается

a. имя класса

b. имя экземпляра объекта

c. ничего не указывается

d. имя сборки

15. Даны описания:

```
class One {
int a; public int b; public static int c;
}
… One x = new One();
```

Укажите правильные способы (один или более) обращения к полям:

a. x.a
b. x.b
c. One.a
d. One.c

16. Члены класса с модификатором доступа private :

a. доступны только внутри класса
b. видны внутри пакета
c. доступны внутри пакета и в классах-наследниках
d. доступны всем

17. Пример использования какого паттерна можно наблюдать в следующем фрагменте кода:

```
FileInputStream in = new FileInputStream("test.dat");
BufferedInputStream bin = new BufferedInputStream(in);
```

a. Stream
b. Wrapper
c. Buffer
d. Decorator
e. Adapter

18. Зачем обычно используют паттерн Flyweight (приспособленец)?

a. Чтобы уменьшить количество создаваемых объектов
b. Чтобы определить интерфейс для создания легковесных объектов
c. Для отделения абстракции от реализации
d. Для приспособления объекта к заданному интерфейсу

19. В чём главное различие паттернов Adapter и Proxy?

a. Adapter создаёт обёртку вокруг одного объекта, Proxy - вокруг множества
b. Эти паттерны нельзя сравнивать
c. Adapter добавляет объекту новый функционал (в отличии от Proxy)
d. Adapter предоставляет другой интерфейс к объекту, Proxy в точности повторяет интерейс объекта

20. Какие два шаблона проектирования из нижеперечисленных используются в следующем фрагменте кода?

```
JButton jbOk = new JButton("OK");

jbOk.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) { System.out.println("OK"); }

});

Action cancelAction = new AbstractAction() {

{ putValue(NAME, "Cancel"); }

public void actionPerformed(ActionEvent e) { System.exit(0); }

};
```

a. Состояние
b. Стратегия
c. Заместитель
d. Декоратор
e. Команда
f. Наблюдатель
g. Прототип

21. Какие из шаблонов относятся к порождающим?

a. Factory
b. Command

c. Strategy
d. Singleton

22. Как называется "интерпретация ссылки на объект как ссылки на базовый тип"?

a. Нисходящее преобразование
b. Восходящее преобразование
c. Свободное преобразование
d. Полиморфизм

23. Общее поведение супер класса может быть передано подклассу с помощью ключевого слова:

a. inherits
b. implements
c. extend
d. implement
e. extends

24. "Черный ящик" как способ повторного использования - это...

a) Стиль повторного использования, основанный на наследовании классов
b) Стиль повторного использования, основанный на композиции объектов.
c) Название структурного паттерна проектирования
d) Стиль повторного использования, основанный на сильной связанности объектов

25. Как в терминах ООП называется объект, который не может быть изменён после создания?

a) Internal object
b) Abstract object
c) Immutable object
d) Sealed object
e) Все варианты
f) Среди вариантов нет правильного ответа

6.2. Контрольные вопросы к экзамену

1. Паттерн Итератор.
2. Противопоставления структурного и объектно-ориентированного анализа.
3. Вопрос первичности функциональной или информационной моделей.
4. Отложенная инициализация.
5. Дайте обзор современных методов программирования.
6. В чем сущность методов процедурного программирования (ПП).
7. Принцип SRP.
8. Параметризованные типы данных
9. Зачем нужны интерфейсы.
10. Абстрактные классы.
11. Функциональные выражения в Java.
12. Композиция и агрегирование.
13. Принцип инверсии зависимостей.
14. Паттерн Стратегия.
15. Дайте обзор стандартных библиотек языка Java.
16. В чем сущность концепции объектно-ориентированного программирования (ООП).
17. Дайте обзор основных понятий ООП.
18. Перечислите преимущества, даваемые инкапсуляцией.
19. Объясните определение класса и объектов класса.
20. Перечислите преимущества, даваемые наследованием.
21. Восходящее и нисходящее преобразование.
22. Объясните особенности простого и множественного наследования.
23. Перечислите преимущества, даваемые полиморфизмом.

24.Паттерн Декоратор.

25.Зачем нужны виртуальные функции и абстрактные классы.

26.Объясните концепцию объектно-ориентированного проектирования.

27.Объясните принцип DRY.

Лист регистрации изменений и дополнений

№ п/п	№ страницы внесения изменений	Дата внесения изменения	Краткое содержание изменений (основание)	Ф.И.О., подпись	«Согласовано» заведующий кафедрой, ведущей дисциплину