ПРИЛОЖЕНИЕ А
ФОНД ОЦЕНОЧНЫХ МАТЕРИАЛОВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
ПО ДИСЦИПЛИНЕ «Программирование»

1. Перечень оценочных средств для компетенций, формируемых в результате освоения дисциплины

Код контролируемой компетенции	Способ оценивания	Оценочное средство
ОПК-2: Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	Курсовой проект; экзамен	Контролирующие материалы для защиты курсового проекта; комплект контролирующих материалов для экзамена
ОПК-4: Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	Курсовой проект; экзамен	Контролирующие материалы для защиты курсового проекта; комплект контролирующих материалов для экзамена
ОПК-5: Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	Курсовой проект; экзамен	Контролирующие материалы для защиты курсового проекта; комплект контролирующих материалов для экзамена
ОПК-6: Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	Курсовой проект; экзамен	Контролирующие материалы для защиты курсового проекта; комплект контролирующих материалов для экзамена

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Оцениваемые компетенции представлены в разделе «Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций» рабочей программы дисциплины «Программирование».

При оценивании сформированности компетенций по дисциплине «Программирование» используется 100-балльная шкала.

Критерий	Оценка по 100-балльной шкале	Оценка по традиционной шкале
Студент освоил изучаемый материал (основной и дополнительный),	75-100	*Отлично*

системно и грамотно излагает его, осуществляет полное и правильное выполнение заданий в соответствии с индикаторами достижения компетенций, способен ответить на дополнительные вопросы.		
Студент освоил изучаемый материал, осуществляет выполнение заданий в соответствии с индикаторами достижения компетенций с непринципиальными ошибками.	50-74	*Хорошо*
Студент демонстрирует освоение только основного материала, при выполнении заданий в соответствии с индикаторами достижения компетенций допускает отдельные ошибки, не способен систематизировать материал и делать выводы.	25-49	*Удовлетворительно*
Студент не освоил основное содержание изучаемого материала, задания в соответствии с индикаторами достижения компетенций не выполнены или выполнены неверно.	<25	*Неудовлетворительно*

3. Типовые контрольные задания или иные материалы, необходимые для оценки уровня достижения компетенций в соответствии с индикаторами

1.Задания на ОПК-2, ОПК-4, ОПК-5, ОПК-6

Компетенция	Индикатор достижения компетенции
ОПК-2 Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.2 Использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1 Применяет стандарты, нормы, правила, техническую документацию в профессиональной деятельности
	ОПК-4.2 Участвует в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК-5 Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1 Инсталлирует программное обеспечение согласно инструкциям
	ОПК-5.2 Коммутирует аппаратное обеспечение в составе информационных и автоматизированных систем
ОПК-6 Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1 Формализует задачу и предлагает алгоритмическое решение
	ОПК-6.2 Проектирует программные продукты с применением основ информатики
	ОПК-6.3 Осуществляет разработку и тестирование программных продуктов

Фонд оценочных материалов по дисциплине «Программирование»

Компетенция	Содержимое компетенции	Индикатор	Содержимое индикатора
ОПК-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	ОПК-2.2	Использует современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1	Применяет стандарты, нормы, правила, техническую документацию в профессиональной деятельности
		ОПК-4.2	Участвует в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК-5	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ОПК-5.1	Инсталлирует программное обеспечение согласно инструкциям
		ОПК-5.2	Коммутирует аппаратное обеспечение в составе информационных и автоматизированных систем
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1	Формализует задачу и предлагает алгоритмическое решение
		ОПК-6.2	Проектирует программные продукты с применением основ информатики
		ОПК-6.3	Осуществляет разработку и тестирование программных продуктов

Компетенция ОПК-2. Индикатор ОПК-2.2

№ 1. ОПК-2.2

Задание. В соответствии с современной методологией структурного программирования и используя язык программирования C/C++, разработать алгоритм и программу для решения следующей задачи профессиональной деятельности.

Задача.

Дан список товаров следующего вида:

1) наименование товара, 2) цена единицы товара, 3) количество единиц товара.

Получить список товаров, цена единицы которых превышает некоторую заданную цену, а количество единиц товара меньше некоторого заданного количества.

№2. ОПК-2.2

Задание. В соответствии с современной методологией структурного программирования и используя язык программирования C/C++, разработать алгоритм и программу для решения следующей задачи профессиональной деятельности.

Задача.

Дан список автомобилей следующего вида:

1) марка автомобиля, 2) цена автомобиля, 3) год выпуска автомобиля.

Получить список автомобилей, цена которых не превышает некоторую заданную цену, а год выпуска – не ранее некоторого заданного года.

№ 3. ОПК-2.2

Задание. На примере проекта простого консольного приложения C++, продемонстрировать возможности распределённой системы управления версиями Git по отмене сделанных в этом файле изменений, работу с ветками (Перемещение по веткам, объединение и удаление. Решение конфликтов слияния). Продемонстрировать работу с удаленным репозиторием Github.com. Показать работу с Git через интерфейс Visual Studio и без него.

№ 4. ОПК-2.2

Задание. Для работы с комплексными числами реализовать на языке java соответствующий класс. Реализовать сложение, вычитание, умножение, деление, модуль,

перегрузку метода equals и toString. Выбрать один из предложенных ниже способов клонирования и реализовать его.

1.Переопределение метода clone() и реализация интерфейса Cloneable();

2.Использование конструктора копирования;

3.Использовать для клонирования механизм сериализации.

Компетенция ОПК-4. Индикатор ОПК-4.1, ОПК-4.2

№1. ОПК-4.1

Задание. В соответствии со стандартами, нормами, правилами и технической документацией, принятыми в профессиональной деятельности, разработать и изобразить в виде структурной схемы алгоритм решения следующей задачи.

Задача.

Дан список студентов факультета в следующем вида:

1) фамилия и инициалы, 2) наименование группы, 3) текущий рейтинг (по 100 балльной системе).

Определить значение самого высокого и самого низкого рейтинга.

№2. ОПК-4.1

Задание. В соответствии со стандартами, нормами, правилами и технической документацией, связанных с профессиональной деятельностью, разработать и оформить код программы для решения следующей задачи.

Задача. Дан одномерный целочисленный массив. Из четных элементов этого заданного массива сформировать другой одномерный массив. Порядок следования элементов в новом массиве сохранить.

№3. ОПК-4.1

Задание. Дан код программы на языке C++. В соответствии со стандартами, нормами, правилами и технической документацией, принятыми в профессиональной деятельности, нарисовать соответствующую UML-диаграмму классов.

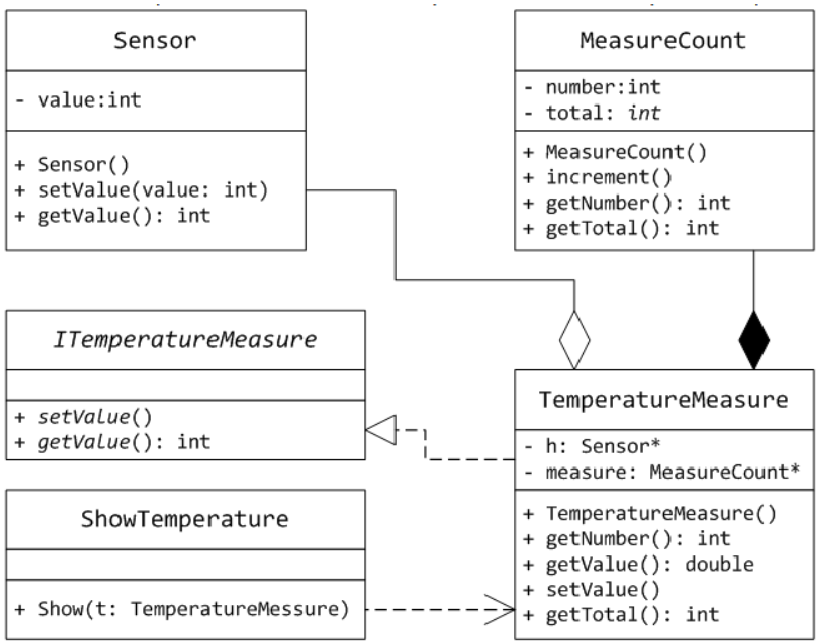
```
#include <iostream>
using namespace std;
class Sensor {
  int value;
```

```
public:
  Sensor() { value = 0; }
  void setValue(int value) { this->value += value; }
  int getValue() { return value; }
};
class MeasureCount
{
  int number;
  static int total;
public:
  MeasureCount() { number = 0; }
  void increment() { number++; total++; }
  int getNumber() { return number; }
  static int getTotal() { return total; }
};
int MeasureCount::total = 0;
class ITemperatureMeasure
{
public:
  virtual void setValue() = 0;
  virtual double getValue() = 0;
};
class TemperatureMeasure : public virtual ITemperatureMeasure
{
private:
  Sensor *h;         // агрегация
  MeasureCount *measure;  // композиция
public:
  TemperatureMeasure(Sensor *h);
  int getNumber() { return measure->getNumber(); }
  double getValue() { return (double)h->getValue() / measure->getNumber(); }
  void setValue()
  {
    int value;
    measure->increment();
    cout << "t[" << measure->getNumber() << "]= ";
    cin >> value;
    h->setValue(value);
  }
  static int getTotal() { return MeasureCount::getTotal(); }
};
TemperatureMeasure::TemperatureMeasure(Sensor *h)
{
  measure = new MeasureCount();
  this->h = h;
}
class ShowTemperature // зависимость
{
public:
  static void Show(TemperatureMeasure t)
  {
    cout << t.getNumber() << ": ";
    cout << t.getValue() << " oC" << endl;
  }
};

int main()
{
  Sensor *h1 = new Sensor();
  TemperatureMeasure tc1(h1);
  for(int i=0; i<5; i++)
    tc1.setValue();
  ShowTemperature::Show(tc1);
  cout << endl;
  Sensor *h2 = new Sensor();
  TemperatureMeasure tc2(h2);
  for (int i = 0; i<5; i++)
    tc2.setValue();
  ShowTemperature::Show(tc2);
  cout << endl;
```

№4. ОПК-4.1

Задание. Дана UML-диаграмма классов. Реализовать на языке C++ представленные на ней классы.



№5. ОПК-4.2

Задание. В процессе разработки технической документации, связанной с созданием программного обеспечения для решения определенной задачи профессиональной деятельности, необходимо оформить отчет о проведенной научно-исследовательской работе. Разработайте и представьте в виде раздела «Содержание» структуру данного отчета.

Задача. Тема: «Разработка программного обеспечения для организации работы со списком анкетных данных». Структура отдельной записи списка:

- ФИО гражданина;

- пол;

- дата рождения (год, месяц, число);

- рост;

- вес.

Необходимо реализовать создание списка, корректировку списка, вывод списка.

№6. ОПК-4.2

Задание. Разработать техническую документацию для кода программы, обеспечив самодокументируемость кода с помощью комментариев. Разработать стандарты, нормы и правила взаимодействия пользователя с данной программой в виде документа «Инструкция для пользователя программы». Программу разработать для решения следующей задачи.

Задача.

Дан список граждан следующего вида:

1) фамилия и инициалы, 2) пол, 3) год рождения.

Получить список граждан мужского пола, имеющих определенный год рождения.

№7. ОПК-4.2

Задание. Вы участвуете в разработке технической документации. Имеется код на C-подобном языке программирования. Вам необходимо нарисовать соответствующую UML-диаграмму классов.

Код программы:

```
class Transformer(){ // базовый класс
    function run(){
        // код, отвечающий за бег
    }
    function fire(){
        // код, отвечающий за стрельбу
    }
}

class Autobot(Transformer){ // дочерний класс, наследование от Transformer
    function transform(){
        // код, отвечающий за трансформацию в автомобиль
    }
}

class Decepticon(Transformer){ // дочерний класс, наследование от Transformer
    function transform(){
        // код, отвечающий за трансформацию в самолет
    }
}

optimus = new Autobot()
megatron = new Decepticon()
```

№8. ОПК-4.2

Задание. Вы участвуете в разработке технической документации. Имеется java-код. Вам необходимо нарисовать соответствующую UML-диаграмму классов.

Код программы:

```
public interface Shape {
    void draw();
```

```
    void erase();
}
public class Circle implements Shape {
    public void draw() {
        System.out.println("Circle.draw()");
    }
}
public class Triangle implements Shape {
    public void draw() {
        System.out.println("Triangle.draw()");
    }
}

public class TestPol {

    public static void main(String[] args) {
        Shape shape1 = new Circle();
        Shape shape2 = new Triangle();
        testPoly(shape1);
        testPoly(shape2);
    }

    public static void testPoly(Shape shape) {
        shape.draw();
    }
}
```

Компетенция ОПК-5. Индикатор ОПК-5.1, ОПК-5.2

№1. ОПК-5.1

Задание. Установить актуальную сборку распределенной системы управления версиями Git и провести ее первоначальную настройку. Настройку провести непосредственным редактированием конфигурационных файлов.

№2. ОПК-5.1

Задание. Установить актуальную версию Java Development Kit (JDK), распространяемую компанией Oracle Corporation, настроить возможность работы с java и javac из командной строки ОС Windows.

№3. ОПК-5.2

Задание. Произвести подключение по USB Android-устройства, его настройку и настройку Windows для работы с Android Debug Bridge (ADB). Продемонстрировать работоспособность ADB, получив список подключенных устройств.

№4. ОПК-5.2

Задание. Продемонстрировать подключение физического устройства (смартфона или планшета) к Android Studio и запуск на нем вашего проекта.

Компетенция ОПК-6. Индикатор ОПК-6.1, ОПК-6.2, ОПК-6.3

№1. ОПК-6.1

Задание. Формализовать задачу и предложить алгоритмическое решение для следующей задачи.

Задача.

Дан список абонентов телефонной сети в следующем виде:

1) фамилия и инициалы, 2) номер телефона, 3) год установки, 4) тип оплаты.

Определить количество абонентов с определенным типом оплаты и сформировать отдельный список таких абонентов.

№2. ОПК-6.1

Задание. Формализовать задачу и предложить алгоритмическое решение для следующей задачи.

Задача.

В модельном агентстве необходимо автоматизировать первый этап отбора претенденток для работы моделью. Исходный список претенденток имеет следующую структуру:

1) ФИО претендентки, 2) рост, 3) обхват талии.

Получить список претенденток для участия во втором этапе отбора по следующему критерию: для участия во втором этапе допускаются девушки, у которых параметры роста и обхвата талии лежат в некоторых заданных диапазонах.

№3. ОПК-6.1

Задание. Формализовать задачу и предложить алгоритмическое решение, используя библиотеку STL, для следующей задачи.

Задача. В двумерном целочисленном массиве определить количество положительных элементов в каждом столбце, а затем выполнить сортировку с предикатом (правило можете выбрать самостоятельно) элементов каждой строки.

№4. ОПК-6.1

Задание. Формализовать задачу и предложить алгоритмическое решение на основе Java Collections Framework для следующей задачи.

Задача. В двумерном целочисленном массиве определить количество нечетных и количество уникальных элементов. Выполнить сортировку с предикатом (правило можете выбрать самостоятельно) элементов каждой строки.

№5. ОПК-6.2

Задание. С применением основ информатики провести проектирование программного продукта для решения следующей задачи.

Задача. Натуральное число из n цифр является числом Армстронга, если сумма его цифр, возведенных в n-ую степень, равна самому числу. Например, $153 = 1^3 + 5^3 + 3^3$. Найти все трехзначные числа Армстронга.

№6. ОПК-6.2

Задание. С применением основ информатики провести проектирование программного продукта для решения следующей задачи.

Задача. Натуральное число называется совершенным, если оно равно сумме всех своих делителей, кроме самого числа. Например, 6 = 1 + 2 +3. Найти все совершенные числа, не превышающие 10000.

№7. ОПК-6.2

Задание. С применением основ информатики, используя язык C#, провести проектирование программного продукта для решения следующей задачи.

Задача. В текстовом файле хранится набор записей. Каждая запись состоит из фамилии студента и номера его группы. Сформировать и распечатать фамилии студентов заданной группы, начинающихся на заданную букву. Фамилия и номер группы вводятся из консоли.

№8. ОПК-6.2

Задание. С применением основ информатики, используя язык java, провести проектирование программного продукта для решения следующей задачи.

Задача. В xml-файле (формат хранения можете выбрать самостоятельно) хранится набор записей. Каждая запись состоит из фамилии студента и номера его группы. Сформировать и распечатать фамилии студентов заданной группы, начинающихся на заданную букву. Фамилия и номер группы вводятся из консоли.

№9. ОПК-6.3

Задание. Осуществить разработку и тестирование программного продукта для решения следующей задачи.

Задача. Гражданин открыл вклад в банке с первоначальной суммой вклада x рублей. Объем вклада каждый год увеличивается на z% по отношению к предыдущему году. Определить, какая сумма вклада будет через n лет.

№10. ОПК-6.3

Задание. Осуществить разработку и тестирование программного продукта для решения следующей задачи.

Задача. Дано натуральное число n. Определить в этом числе номер некоторой заданной цифры, считая от конца числа. Если такой цифры в числе нет, ответом должно быть число 0. Если таких цифр в числе несколько, должен быть определен номер самой правой из них.

№11. ОПК-6.3

Задание. Осуществить разработку на языке C# и тестирование программного продукта для решения следующей задачи.

Задача. Вводится целочисленная матрица и целое число. Элементы этой матрицы, без остатка делящиеся на введенное число, переписать в массив и найти их произведение. Напечатать матрицу, массив и найденное произведение.

№12. ОПК-6.3

Задание. Осуществить разработку на языке Java и тестирование программного продукта для решения следующей задачи.

Задача. Задан массив (ArrayList) целых чисел, в котором содержатся элементы Integer от 1 до N. В этом массиве есть один и только один продублированный элемент.

1) Создать метод для генерации такого массива;
2) Создать метод для нахождения этого продублированного элемента.

4. Файл и/или БТЗ с полным комплектом оценочных материалов прилагается.