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 **Основы объектно-ориентированного программирования**

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 3

|  |  |
| --- | --- |
| Форма аттестации | Семестр |
| Экзамен | 3 |

Новосибирск 2019

**Фонд оценочных средств** промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Основы объектно-ориентированного программирования», 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, протокол № 75 от 02.07.2019.

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 «Основы объектно-ориентированного программирования» 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 3 | |
| Портфолио | Экзамен |
| **ОПК-1**: Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности, в части следующих результатов обучения: | | | |
| ОПК-1.1 | Знать: основы  математики, физики, вычислительной техники и программирования | + | + |
| ОПК-1.2 | Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования | + | + |
| ОПК-1.3 | Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности | + |  |
| **ОПК-8:** Способен разрабатывать алгоритмы и программы, пригодные для практического применения, в части следующих результатов обучения: | | | |
| ОПК-8.1 | Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | + | + |
| ОПК-8.2 | Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули | + |  |
| ОПК-8.3 | Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | + |  |

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

Категория 2 – Средства объектно-ориентированного программирования C++

* Средства поддержки процедурного стиля в языке C++
* Средства поддержки объектно-ориентированного подхода в С++
* Средства поддержки обобщенного программирования в С++
* Основы STL и стандартная библиотека C++
  1. Порядок проведения промежуточной аттестации по дисциплине

Промежуточная аттестация проводится в форме экзамена и включает 2 этапа: портфолио и экзамен. Необходимым условием для прохождения промежуточной аттестации является оценка «зачтено» по результатам выполненного портфолио. Для оценивания портфолио студенту необходимо сдать все работы, входящие в структуру портфолио.

Оценка 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 при типичной стоимости задачи от 10 до 30 баллов. Должно быть сдано не менее 4 заданий.

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 3 | | | |
| 1 | Коллоквиум | Средство контроля усвоения учебного материала дисциплины, организованное как письменный тест | Вопросы по темам/разделам дисциплины |
| 2 | Портфолио | Целевая подборка работ студента, раскрывающая его индивидуальные образовательные достижения в одной или нескольких учебных дисциплинах. | Структура портфолио |
| 3 | Экзаменационный билет | Комплекс вопросов и/или разноуровневых заданий (списать подходящее из п. про задачи) | Список теоретических вопросов и задач |

* 1. Требования к структуре и содержанию оценочных средств  
     аттестации
     1. **Вопросы коллоквиума**

1) Сформулируйте основную идею объектно-ориентированного программирования. В чем его преимущества перед процедурным стилем?

2) В чем состоит принцип абстракция в ООП? Какие средства языка C++ поддерживают данный принцип?

3) Что такое инкапсуляция? Какие ключевые слова C++ относятся к инкапсуляции?

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

5) Определите понятие класса. Перечислите варианты отношений между классами. Обозначьте взаимосвязь между этими отношениями, если она существует. Какое из отношений является наиболее сильным по степени зависимости между классами?

6) Что такое абстрактные классы и интерфейсы? Для чего они нужны в ООП?

7) В приведенном классе перечислите (отметьте) переменные и методы, обеспечивающие состояние и поведение объектов класса Detail:

class Detail {

const int serial;

Detail \*pcomponents;

double weight;

static const int ARTICLE;

protected:

Detail();

Detail(const Detail&);

public:

Detail (int);

static void performAction();

int getSerial() const;

double getWeight() const;

void doSomething();

~Detail();

};

8) В приведенном примере класса MobilePhone, отметьте, какие методы являются конструкторами, деструкторами, модификаторами и селекторами.

class MobilePhone {

//…

public:

MobilePhone (long networkId = 0);

MobilePhone (const MobilePhone &);

const MobilePhone & operator=(const MobilePhone &);

int getOwnNumber() const;

int popLastCall();

void redial();

void placeCall(int number);

friend ostream& operator<<(ostream&, const MobilePhone&);

~MobilePhone ();

static Network\* getAllKnownNetworks();

};

9) Перечислите примитивные типы языка С++. Что вы можете отметить об относительном размере (sizeof) этих типов по отношению друг к другу?

10) Напишите следующие объявления (переменных): указатель на булеву переменную, указатель на константный long, массив из 10 объектов string, указатель на указатель на символ, константный указатель на целое , ссылка на float, ссылка на константный объект типа string. Проинициализируйте эти переменные значениями, используя определения.

11) Напишите функцию, которая обменивает значения двух чисел типа double. В качестве типа аргументов воспользуйтесь указателями на double. Напишите аналогичную функцию, использующую ссылочные аргументы.

12) Перечислите классы памяти в программах написанных на языке С++.

13) Отметьте, в какой памяти будет выделено пространство под каждую из перечисленных переменных (ее содержимое):

struct Record {

static const int MAX\_VALUE=0x7FFFFFFF; //MAX\_VALUE?

int value;

const int constValue;

static double staticValue;

Record(): constValue(0) {}

static void foo();

};

Record object; //object?

double Record::staticValue = 1.2; //staticValue?

void Record::foo() {

static int counter=1; //counter?

Record o; //o.value? o.constValue?

Record &r = o; //r.value? r.constValue?

}

const double PI= 3.1415926; //PI?

int main (int argc, char\*\* argv) { //argc?

Record &ref = object; //ref? ref.staticValue?

string s[10]; //s[1] ?

Record \*ptr = &ref; //ptr->value?

int i =10; //i?

Record \*pobj = new Record[10]; //pobj? \*pobj? pobj[5]?

}

14) Что распечатает следующая программа на С++, какое место в программе Вам кажется потенциально опасным и почему?

#include <iostream>

using namespace std;

class A{

public:

A(){

cout<<"+";

}

A(const A& a){

cout<<"#";

}

~A(){

cout<<"-";

}

};

class B{

A a;

public:

B(A tmp){

cout <<"B";

a = tmp;

}

A& getA() {

return a;

}

};

A global;

int main(int argc, char \*\*argv) {

cout<<"begin";

A \*pa = new A();

A local(\*pa);

B b(global);

local = b.getA();

cout<<"finish";

}

15) Что распечатает следующая программа на С++ (в предположении что calloc отработает успешно), дополнительно укажите что в этой программе на Ваш взгляд неправильно/некрасиво и почему?:

#include <iostream>

#include <stdlib>

using namespace std;

class A{

static int counter;

int i;

public:

A() {

i = ++counter;

}

~A() {

counter--;

}

friend ostream& operator<<(ostream &, const A&);

};

ostream& operator<<(ostream & out, const A& a) {

return out<<a.i;

}

int A::counter = 0;

A global;

int main(int argc, char \*\*argv) {

cout<<global<<endl;

A \*p2A = static\_cast<A\*> ( calloc(5, sizeof(A)) );

A localArray[5];

A \*pA = new A[3];

cout<< p2A[2] << endl;

cout<< localArray[3] << endl;

cout<< \*pA << endl;

delete[] pA;

free (p2A);

A local;

cout<< local << endl;

}

16) Для класса Widget задайте констркутор (агументы определите самостоятельно), конструктор копии, деструктор и оператор присваивания, при условии что классы Detail и Specification полностью реализованы:

class Widget {

private:

const int serial; //серийный номер

double weight;

Specification \*pspec; //указатель на спецификацию (ассоциация)

int detailsCount; //число компонент

Detail \*pdetail; //компоненты (композиция)

};

**2.1.2. Требования к портфолио**

Портфолио должно содержать результаты участия в коллоквиуме (письменный тест) и 4-6 выполненных заданий по следующим темам:

* Раздельная компиляция и пространства имен, структурные средства языка C++
* Перегрузка функций, указатели на функции, перечисления
* Классы в языке C++
* Иерархии классов, наследование
* Обобщенное программирование

**2.1.3. Форма и перечень вопросов экзаменационного билета**

**Форма  экзаменационного билета**

Таблица П1.3

|  |  |  |  |
| --- | --- | --- | --- |
|  | Новосибирский государственный университет  **Экзамен** | |  |
|  | Основы объектно-ориентированного программирования | |  |
|  | 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** | |
| ОПК -1.1 | Эволюция методологий программирования. Парадигмы программирования |
| ОПК -1.1 | Основные принципы объектного подхода. Абстрагирование |
| ОПК -1.1 | Основные принципы объектного подхода. Инкапсуляция |
| ОПК -1.1 | Основные принципы объектного подхода. Модульность |
| ОПК -1.1 | Основные принципы объектного подхода. Иерархия |
| ОПК -1.1 | Основные принципы объектного подхода. Типизация |
| ОПК -1.1 | Объект с точки зрения ООП. Состояние. Поведение |
| ОПК -1.1 | Объект с точки зрения ООП. Идентичность и жизненный цикл объектов |
| ОПК -1.1 | Объект с точки зрения ООП. Взаимоотношения между объектами. |
| ОПК -1.1 | Классы. Природа классов. Метамодель. Инстанцирование. |
| ОПК -1.1 | Классы. Структура класса. Абстрактные классы и интерфейсы |
| ОПК -1.1 | Классы. Принцип подстановки Лисковой. Принцип разделения интерфейсов |
| ОПК -1.1, ОПК-1.2 | Классы. Средства UML для построения диаграмм классов |
| ОПК -1.1, ОПК-1.2 | Классы. Отношения между классами. Ассоциация и агрегация |
| ОПК -1.1, ОПК-1.2 | Классы. Иерархии классов. Зависимость |
| **Категория 2 – Средства объектно-ориентированного программирования на C++** | |
| ОПК-8.1 | Модель памяти и структура программы. Классы памяти. Ссылки |
| ОПК-8.1 | Средства абстракции C++. Структура класса. Статические члены и их инициализация |
| ОПК-8.1 | Средства инкапсуляции C++. Инкапсуляция и наследование. Друзья |
| ОПК-8.1 | Модульность, раздельная компиляция, пространства имен, using директива. |
| ОПК-8.1 | Представление иерархических отношений. Наследование |
| ОПК-8.1 | Представление иерархических отношений. Агрегация. Зависимость по времени жизни |
| ОПК-8.1 | Правила преобразования типов в C++. Параметрический и виртуальный полиморфизм |
| ОПК-8.1 | C++: средства реализации состояния объектов; реализация поведения |
| ОПК-8.1 | Перегрузка операторов |
| ОПК-8.1 | Жизненный цикл объекта. Инициализация массивов. Конструкторы и деструкторы. Порядок вызова конструкторов и деструкторов при наследовании |
| ОПК-8.1 | Варианты реализации отношения клиент-сервер. Объекты при передаче параметров и возврате из методов |
| ОПК-8.1 | Исключения в C++. Обработка исключений. Умные указатели |
| ОПК-8.1 | Шаблоны классов и шаблоны функций. Специализация |
| ОПК-8.1 | Основы STL. Структура и назначение. Контейнеры |
| ОПК-8.1 | Основы STL. Аллокаторы и итераторы |

Набор экзаменационных билетов формируется и утверждается в установленном порядке в начале учебного года при наличии контингента обучающихся, завершающих освоение дисциплины «Основы объектно-ориентированного программирования» 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») |
| ОПК-1 | Портфолио,  Экз. вопрос категории 1 | ОПК-1.1 Знать:основы  математики, физики, вычислительной техники и программирования | Не знает нотации языка моделирования UML в части диаграмм классов, диаграмм последовательности | Знает основные элементы нотации, может читать диаграммы | Знает все основные элементы языка UML для моделирования отношений между классами их назначение | Знает особенности применения элементов нотации UML |
| ОПК-1 | Портфолио, Экз. вопрос категории 1 | ОПК- 1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и общеинженерных знаний, методов математического анализа и моделирования | 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 и агрегации и композиции | Уверенно строит диаграммы классов , решения практических задания портфолио снабжены диаграммами демонстрирующими структуру программы |
| ОПК-1 | Портфолио | ОПК-1.3 Владеть: навыками теоретического и экспериментального исследования объектов профессиональной деятельности | Не знает какие основные средства используются на разных платформах (Windows, Linux, Mac OS X). Не умеет пользоваться средой разработки | Знает основные характеристики сред для разных платформ. Умеет использовать основные функции для редактирования одномодульной программы и компиляции | Знает характеристики сред для разных платформ, Демонстрирует уверенные знания функций одной из сред разработки. Умеет пользоваться средой разработки для реализации многомодульных приложений владеет средствами отладки | Демонстрирует глубокие познания в достоинствах и недостатках основных средств разработки для разных платформ, уровень поддержки ими современного стандарта языка. Умеет пользоваться средой разработки, разрабатывать многомодульные приложения, пользоваться полным спектром средств оптимизации и отладки |
| ОПК-8 | Портфолио, Экз. вопрос категории 2 | ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения | Имеет фрагментарные знания в области структурных средств разработки C++, не знает правил перегрузки операций и функций | Знает основные алгоритмические и структурные средства языка, основные типы данных и модель сборки | Демонстрирует уверенные знания основных элементов и структурных средств языка, типов данных правил разрешения параметрического полиморфизма | Демонстрирует углубленные знания основных алгоритмических, функциональных и процедурных средств, знает особенности оптимизации на уровне языка ассемблера |
| ОПК-8.2 | Портфолио  (сдача заданий) | ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули | не умеет писать тесты с целью верификации корректности реализованных программ с использованием библиотеки Google Test | Умеет разрабатывать простейшие тесты с использованием технологии Google Test | Уверенно пишет тесты с использованием технологии Google Test. Степень тестового покрытия программ практического задания не менее 50% | Пользуется продвинутыми средствами технологии Google Test, тестовое покрытие программ практических заданий не менее 75% |
| ОПК-8.3 | Портфолио (сдача заданий) | ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы | Не умеет реализовать алгоритмы и определять собственные типы данных (классы) | Умеет реализовать основные алгоритмы с использованием простых средств языка, может определять несложные классы | Реализует алгоритмы, разрабатывает систему классов, может реализовать простые шаблоны и обобщенные алгоритмы | Владеет всеми средствами языка, может программировать сложные шаблоны с переменным числом аргументов |

# Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в 3 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.

Итоговая оценка результатов промежуточной аттестации выставляется по следующей формуле:

Итоговая Оценка = 0.4 • О\_1 + 0.6 • О\_2;

О\_1 - итоговая оценка по компетенциям, не вынесенным экзамен,

О\_2 - итоговая оценка по компетенциям, вынесенным на экзамен.

Оценки О\_1 и О\_2 представляют из себя соответствующие средние арифметические оценок по компетенциям не вынесенным и вынесенным соответственно на экзамен.

**Лист актуализации фонда оценочных средств промежуточной аттестации**

**по дисциплине  
«Основы объектно-ориентированного программирования»**

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