Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Кислова Наталья Николаевна
Должность: Проректор по УМР и качеству образования
Дата подписания: 06.06.2025 12:29:57
Уникальный программный ключ:
52802513f5b14a975b3e9b13008093d5726b159bf6064f865ae65b96a966c035

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Самарский государственный социально-педагогический университет»
Кафедра информатики, прикладной математики и методики их преподавания

Утверждаю
Проректор по учебно-методической
работе и качеству образования
_____ Н.Н. Кислова

Пугач Ольга Исааковна, Петрунин Дмитрий Андреевич

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации по дисциплине
«Программирование»

Направление подготовки:
44.03.01 Педагогическое образование
Направленность (профиль):
«Информатика»

Квалификация выпускника
Бакалавр

Рассмотрено
Протокол №10 от 27.05.2025
Заседания кафедры информатики, прикладной математики и методики их преподавания

Одобрено
Начальник Управления образовательных программ
Н.А. Доманина

Самара 2025

Пояснительная записка

Фонд оценочных средств (далее – ФОС) для промежуточной аттестации по дисциплине «Программирование» разработан в соответствии с федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 44.03.01 Педагогическое образование. Утвержден приказом Министерства образования и науки Российской Федерации от 22 февраля 2018 г. № 121 основной профессиональной образовательной программой высшего образования «Информатика» с учетом требований профессионального стандарта 01.001 «Педагог (педагогическая деятельность в сфере дошкольного, начального общего, основного общего, среднего общего образования) (воспитатель, учитель)», утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 18 октября 2013 г. № 544н (зарегистрирован Министерством юстиции Российской Федерации 6 декабря 2013 г., регистрационный № 30550), с изменениями, внесенными приказами Министерства труда и социальной защиты Российской Федерации от 25 декабря 2014 г. № 1115н (зарегистрирован Министерством юстиции Российской Федерации 19 февраля 2015 г., регистрационный № 36091) и от 5 августа 2016 г. № 422н (зарегистрирован Министерством юстиции Российской Федерации 23 августа 2016 г., регистрационный № 43326), 01.003 «Педагог дополнительного образования детей и взрослых» утвержденный приказом Министерства труда и социальной защиты Российской Федерации от 22 сентября 2012 г. № 652н от 22.09.2021 г. (Зарегистрировано в Минюсте России 17.12.2021 N 66403).

Цель ФОС для промежуточной аттестации – установление уровня сформированности части компетенции – ПК-1.

Задачи ФОС для промежуточной аттестации - контроль качества и уровня достижения результатов обучения по формируемым в соответствии с учебным планом компетенциям:

ПК-1. Способен осуществлять педагогическую деятельность по реализации образовательного процесса по предмету

ПК-1.1. Умеет реализовывать образовательную программу по предмету с учетом специфики содержания, методов и инструментов соответствующей области научного знания

Знает: современные методы процедурного, объектно-ориентированного, функционального программирования; основные конструкции языков программирования; основные типы данных и операторы.

Умеет: записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; подбирать оптимальное программное обеспечение для разработки, тестирования и отладки программ.

ПК-1.2. Реализует образовательную программу по предмету с использованием технологий профессиональной деятельности

Умеет: использовать знания по программированию при реализации образовательной программы по информатике.

Требование к процедуре оценки:

Помещение: особых требований нет

Расходные материалы: лист бумаги, ручка

Доступ к дополнительным справочным материалам: не предусмотрен

Нормы времени: 130 мин

Комплект оценочных средств для проведения промежуточной аттестации в 7 семестре

Раздел 1. Процедурное программирование на C++

Проверяемая компетенция:
ПК-1. Способен осуществлять педагогическую деятельность по реализации образовательного процесса по предмету
Проверяемый индикатор достижения компетенции:

ПК-1.1. Умеет реализовывать образовательную программу по предмету с учетом специфики содержания, методов и инструментов соответствующей области научного знания
Проверяемые результаты обучения:
Знает: современные методы процедурного, объектно-ориентированного, функционального программирования; основные конструкции языков программирования; основные типы данных и операторы.
Умеет: записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; подбирать оптимальное программное обеспечение для разработки, тестирования и отладки программ.
Проверяемый индикатор достижения компетенции:
ПК-1.2. Реализует образовательную программу по предмету с использованием технологий профессиональной деятельности

Проверяемые результаты обучения:
Умеет: использовать знания по программированию при реализации образовательной программы по информатике.

Тип (форма) задания: тестовые задания с выбором одного или нескольких вариантов ответа (A), практические задания (B).

Содержание задания:

A1. Выберите правильный вариант объявления константной переменной в C++, где type - тип данных в C++ variable - имя переменной value - константное значение

a) const variable = value;
b) const type variable = value;
c) const type variable := value;

A2. Чтобы подключить заголовочный файл в программу на C++, например iostream необходимо написать:

a) #include <> с iostream внутри скобок
b) #include <>; с iostream.h внутри скобок
c) include #iostream,h;
d) include (iostreamh)

A3. Какой из ниже перечисленных операторов, не является циклом в C++?

a) while
b) repeat until
c) for
d) do while

A4. Чему будет равна переменная a, после выполнения этого кода int a; for(a = 0; a < 10; a++) {}?

a) 9
b) 10
c) 1
d) 17

A5. Укажите объектно-ориентированный язык программирования

a) Все варианты ответов
b) C++
c) Java
d) Eiffel

A6. Какую функцию должны содержать все программы на C++?

a) system()
b) start()
c) main()
d) program()

A7. Какие служебные символы используются для обозначения начала и конца блока кода?
a) { }
b) < >
c) begin end
d) ( )

A8. Цикл с постусловием?
a) while
b) for
c) do while
d) repeat

A9. Какая из следующих записей - правильный комментарий в C++?
a) */ Комментарии */
b) {комментарий}
c) /* комментарий */
d) ** Комментарий **

A10. Программа, переводящая входную программу на исходном языке в эквивалентную ей выходную программу на результирующем языке, называется:
a) интерпретатор
b) сканер
c) транслятор
d) компилятор

A11. Цикл с предусловием?
a) while
b) do while
c) for
d) repeat

A12. Какой оператор не допускает перехода от одного константного выражения к другому?
a) Stop;
b) end;
c) точка с запятой
d) break;

A13. Какими знаками заканчивается большинство строк кода в Си++?
a) ; (точка с запятой)
b) : (двоеточие)
c) , (запятая)
d) . (точка)

A14. Каков результат работы следующего фрагмента кода?

```
int x = 0;

  switch(x)
  {

    case 1: cout << "Один";

    case 0: cout << "Нуль";

    case 2: cout << "Привет мир";

  }
```

a) НульПривет мир
b) Привет мир
c) Нуль
d) Один

A15. Какое значение, по умолчанию, возвращает программа операционной системе в случае успешного завершения?

a) 0
b) 1
c) -1
d) Программа не возвращает значение.

A16. Простые типы данных в C++.
a) целые – int, вещественные – float или double, символьные – string
b) целые – int, вещественные – float или double, символьные – char
c) целые – int, вещественные – float или real, символьные – char
d) целые – bool, вещественные – float или double, символьные – string

A17. Какой служебный знак ставится после оператора case ?
a) :
b) ;
c) .
d) -

18. Укажите правильное определение функции main в соответствии со спецификацией стандарта ANSI
a) int main(void)
b) int main()
c) void main()
d) void main(void)

19. Какому зарезервированному слову программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением?
a) default
b) all
c) other
d) contingency

20. Какие среды программирования (IDE) предназначены для разработки программных средств?
a) MVS, NetBeans, QT Creator, RAD Studio, Dev-C++
b) MVS, Code::Blocks, QT Creator, RAD Studio, MathCAD
c) MVS, Code::Blocks, QT Creator, AutoCAD, Eclipse

Оценочный лист к типовому заданию A

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
a	a	b	a	a	c	a	c	c	c
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
a	d	a	a	a	b	a	a	a	a

Каждый верный ответ оценивается в 2 балла.

Задание B

B1. Оцените следующий алгоритм. Что вернет программа при вводе num5. Опишите, что делает программа в общем случае? Предложите альтернативный вариант решения.

```
#include <iostream>
#include <locale.h>

using namespace std;

int main()
{
  int number;
  setlocale(LC_CTYPE,"Russian");
  cout << "Введите число: ";
  cin >> number;
  cin.ignore();
  cout << "Вы ввели: "<< number <<"\n";
  cin.get();
}
```

B2. Последовательность состоит из натуральных чисел и завершается числом 0. Всего вводится не более 10000 чисел (не считая завершающего числа 0). Определите, сколько элементов этой последовательности равны ее наибольшему элементу.

Оценочный лист к типовому заданию (модельный ответ):

B1. Программа вернет 5. Программа возвращает введенное число, игнорируя остальные символы. Засчитывается любое альтернативное работоспособное решение. Задача оценивается в 10 баллов

B2. Оценивается работоспособность программы и скорость работы алгоритма. Задача оценивается в 10 баллов

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
ПК-1.1	Задание А 1-5	40	40	20-24	28-32	36-40
ПК-1.2	Задание В	20	20	10-14	15-16	17-20

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.

Вариант 2

A1. Оператор вывода cout может печатать несколько значений или переменных в одной команде, используя следующий синтаксис:

a) cout << "Привет", name, "n";
b) cout << "Привет" + name + "n";
c) cout << ("Привет" & name & "n");
d) cout << "Привет" << name << "n";

A2. Какое значение будет напечатано, в результате выполнения программы?

```
#include
 int main()
{
 int x = 3;
  switch(x)
 {
  case 0:
   int x = 1;
   std::cout << x << std::endl;
  break;
  case 3:
   std::cout << x << std::endl;
  break;
  default:
   x = 2;
   std::cout << x << std::endl;
 }
  return 0;
```

}

a) 3
b) 2
c) ничего не напечатается, программа вообще не будет работать
d) 1
e) 0

A3. Тело оператора выбора if, будет выполняться, если его условие:

a) ложно (false)
b) истинно (true)
c) выполнение не зависит от условия

A4. Укажите блок кода, в котором переменная y доступна.

```
int main(int argc, char** argv)
{
 if ( argc > 10 )
{
}
 else if (int y = argc - 1 )
 {
   }
 else
 {
 }
 return 0;
 }
```

a) строки 8 -11
b) строки 8 -17
c) строки 8 -15
d) строки 4 -17
e) строки 4 -15

A5. Что появится на экране, после выполнения этого фрагмента кода?

```
int a = 1, b =2;
if (a == b);
cout << a << " = " << b << endl;
```

a) синтаксическая ошибка
b) 1 = 2
c) a = b
d) вывод на экран не выполнится

A6. Результат выполнения следующего фрагмента кода: !((1 || 0) && 0)

a) результат не может быть заранее определен
b) 1
c) 0
d) 4

A7. Какое из следующих значений эквивалентно зарезервированному слову true?

a) 0.1
b) 1
c) -1
d) 66
e) Все варианты ответов

A8. Это значение 5.9875e17 может быть сохранено в переменной, типа

a) Bool
b) Float
c) Int
d) Long
e) Short

A9. Вывод данных в C++

a) cout « <переменная >,< “< строка выводится на экран>” ,<выражение > ,endl;
b) cout « <переменная >,< “< строка выводится на экран>” ,<выражение > ,endl;

c) cout « <переменная > « “< строка выводится на экран>” « <выражение > « endl;

A10. В каком случае лучше всего использовать приведение типов данных?
a) во всех выше указанных случаях
b) чтобы разрешить программе использовать только целые числа
c) чтобы изменить тип возвращаемого значения функции
d) при делении двух целых чисел, для того, чтобы вернуть результат с плавающей точкой

A11. Какой тип данных имеет переменная ARGV?
a) это не переменная
b) char **
c) char *
d) int

A12. Что будет напечатано на экране, после выполнения этого кода?
```
#include
 int foo(int y);
int foo(int x)
{
  return x+1;
}
 int main(int argc, char** argv)
{
  int x = 3;
  int y = 6;
   std::cout << foo(x) << std::endl;
   return 0;
}
```
a) 4
b) ошибка компиляции
c) 3
d) 9

A13. Какая строка содержит зарезервированные слова языка программирования C++?
a) sizeof, const, typedef, static, voided, enum, struct, union
b) char, int, float, doubled, short, long, unsigned, signed
c) if, else, for, while do, switch, continue, break
d) defaulted, goto, return, extern, private, public, protected

A14. Что будет напечатано, после выполнения этого кода: cout << (5 << 3); ?
a) 35
b) 53
c) 40
d) 56

A15. Укажите неправильно записанную операцию отношения
a) >=
b) <=
c) =!
d) все операторы записаны правильно

A16. Результат выполнения следующего фрагмента кода: cout << 22 / 5 * 3;
a) 12
b) 13.2
c) 1
d) 1.47
e) Другое

A17. В каком случае можно не использовать фигурные скобочки в операторе выбора if?
a) если в теле оператора if всего один оператор
b) если в теле оператора if два и более операторов
c) нет правильного ответа
d) если в теле оператора if нет ни одного оператора

A18. Ввод данных в C++

a) cin » <выражение1> » <выражение2>…;
b) cin » <выражение1>,<выражение2>,…;
c) cin » <выражение1> » <выражение2> » endl »…;

A19. Какое ключевое слово указывает, что целая переменная не может принимать отрицательные значения?
a) Unsigned
b) Positive
c) нет такого зарезервированного слова
d) long
e) другое

A20. Преобразование целочисленной переменной value в ASCII эквивалент
a) cout << value
b) atoi(value)
c) (char) value
d) char (value)

Оценочный лист к типовому заданию А

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
d	c	b	c	b	b	e	b	c	d
A11	A12	A13	A14	A15	A16	A17	A18	A19	A20
b	b	c	c	c	b	a	a	a	c,d

Каждый верный ответ оценивается в 2 балла.

Задание В

B1. Оцените следующий алгоритм. Что вернет программа при вводе числа 105. Опишите, что делает программа в общем случае? Предложите альтернативный вариант решения.

```
void prime_divisors(int n)
{
    int t,k;
    cout<<n<<" =";
    if (n<=1)
    {
            cout<<" "<<n;
            return ;
    }
    t=sqrt(n);
    k=2;
    while (k<=t)
    {
            if (!(n%k))
            {
                    n/=k;
                    cout<<" "<<k;
                    while (!n%k)
                    {
                            n/=k;
                            cout<<" "<<k;
                }
                t=sqrt(n);
            }
            k++;
     }
     if (n>1)
            cout<<" "<<n<<endl;
}
```

B2. На электронную почту Вам пришло письмо, подписанное аббревиатурой (первыми буквами фамилии, имени и отчества (далее - ФИО) отправителя). Аббревиатура оказалась Вам незнакома. У Вас есть список всех предполагаемых отправителей, взятый из ранее полученных писем, среди которых различных людей с такой аббревиатурой не больше 10.
Вам предлагается написать эффективную, в том числе по используемой памяти, программу, которая определит всех вероятных адресатов – людей, ФИО которых можно сократить до нужной аббревиатуры. ФИО следует

выдать в порядке убывания частоты их встречаемости в списке, если несколько человек встречаются с одинаковой частотой, то вывести их в алфавитном порядке.

Оценочный лист к типовому заданию (модельный ответ):

B1. Программа вернет 3*5*7. Программа раскалывает введенное число на простые сомножители. Засчитывается любое альтернативное работоспособное решение. Задача оценивается в 10 баллов

B2. Оценивается работоспособность программы и скорость работы алгоритма. Задача оценивается в 10 баллов

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
ПК-1.1	Задание А 1-20	40	40	20-24	28-32	36-40
ПК-1.2	Задание В	20	20	10-14	15-16	17-20

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.

Проверяемая компетенция:
ПК-1. Способен осуществлять педагогическую деятельность по реализации образовательного процесса по предмету
Проверяемый индикатор достижения компетенции:

ПК-1.1. Умеет реализовывать образовательную программу по предмету с учетом специфики содержания, методов и инструментов соответствующей области научного знания
Проверяемые результаты обучения:
Знает: современные методы процедурного, объектно-ориентированного, функционального программирования; основные конструкции языков программирования; основные типы данных и операторы.
Умеет: записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; подбирать оптимальное программное обеспечение для разработки, тестирования и отладки программ.
Проверяемый индикатор достижения компетенции:
ПК-1.2. Реализует образовательную программу по предмету с использованием технологий профессиональной деятельности

Проверяемые результаты обучения:
Умеет: использовать знания по программированию при реализации образовательной программы по информатике.

Комплект оценочных средств для проведения промежуточной аттестации в 6 семестре

Раздел 2. Объектно-ориентированное программирование (Java)

1. Which of the following statements are true? (Отметьте все варианты)
    1. Runtime exceptions are the same thing as checked exceptions.
    2. Runtime exceptions are the same thing as unchecked exceptions.
    3. You can declare only checked exceptions.
    4. You can declare only unchecked exceptions.
    5. You can handle only Exception subclasses.
2. Which of the following pairs fill in the blanks to make this code compile? (Choose all that apply)

```
7: public void ohNo() _____ Exception {
8: _____________ Exception();
9: }
```

    1. On line 7, fill in throw
    2. On line 7, fill in throws
    3. On line 8, fill in throw
    4. On line 8, fill in throw new
    5. On line 8, fill in throws
    6. On line 8, fill in throws new
3. When are you required to use a finally block in a regular try statement (not a try-withresources)?
    1. Never.
    2. When the program code doesn’t terminate on its own.
    3. When there are no catch blocks in a try statement.
    4. When there is exactly one catch block in a try statement.
    5. When there are two or more catch blocks in a try statement.
4. Which exception will the following throw?

```
Object obj = new Integer(3);
String str = (String) obj;
System.out.println(str);
```

    1. ArrayIndexOutOfBoundsException
    2. ClassCastException
    3. IllegalArgumentException
    4. NumberFormatException
    5. None of the above.
5. Какие утверждения верны? (Отметьте все варианты)
    1. Checked exceptions are allowed to be handled or declared.
    2. Checked exceptions are required to be handled or declared.
    3. Errors are allowed to be handled or declared.
    4. Errors are required to be handled or declared.
    5. Runtime exceptions are allowed to be handled or declared.
    6. Runtime exceptions are required to be handled or declared.

6. Тест, который проверяет, что функция деления при запросе 2/0 вернет ArithmeticException, это:
    1. Unit тест
    2. нагрузочный тест
    3. функциональный тест
7. Любой поток – это объект?
    1. да
    2. нет
8. Аннотация @Deprecated на методе класса означает:
    1. что метод устарел и его больше не поддерживают
    2. что в методе есть ошибка
9. Каждый класс потока для чтения данных из конкретного устройства унаследован от InputStream?
    1. да
    2. нет
10. Синхронное обращение к базе означает:
    1. приложение передает в базу callback, который будет вызван базой после выполнения запроса
    2. приложение ждет, пока будет закрыто соединение к базе
11. Метод, проаннотированный @Before, будет запущен:
    1. один раз перед стартом всех тестов
    2. один раз при запуске тестов класса, в котором он написан
    3. при запуске каждого метода, аннотированного как @Test
12. Какие варианты решения проблемы взаимодействия потоков могут быть расширены на взаимодействие процессов (запущенных на разных компьютерах)?
    1. держать в каждом потоке ссылки на все остальные потоки
    2. менять состояние общего для потоков объекта
    3. передавать сообщения через общий ресурс
    4. передавать сообщения через систему обмена сообщениями
13. Сборщик мусора это:
    1. часть JVM, которая удаляет объекты, на которые нет ссылок
    2. часть перекомпайлера Java, который расставляет операторы delete
    3. часть JVM, которая удаляет переменные на стеке, которые вышли за область видимости
    4. часть JVM, которая удаляет объекты, по ссылкам на которые нельзя дойти до стека
14. Будет ли сериализовано статическое поле?
    1. да
    2. нет
    3. да, если его отметить ключевым словом transient
15. Тест, который проверяет, что два пользователя могут зайти в игру и начать партию в шахматы:
    1. нагрузочные тесты
    2. Unit тесты
    3. функциональные тесты
16. Какие утверждения относительно классов WindowListener и WindowEvent верны?
    1. данные классы можно использовать для отслеживания изменения количества компонент в контейнере
    2. событие WindowListener отслеживает нажатие клавиш клавиатуры
    3. событие WindowEvent сигнализирует об изменении состояния окна
    4. данные классы можно использовать для инициирования закрытия окна приложения
17. Для чего нужны классы-обертки?
    1. использование классов-оберток ускоряет работу приложения
    2. переменная класса-обертки может принимать значения null
    3. классы-обертки предоставляют набор статических методов для работы с примитивными типами
    4. при использовании коллекций нельзя использовать значения примитивных типов
18. Почему объектно-ориентированное программирование пришло на смену процедурному программированию?
    1. использование ООП позволяет создавать не зависящие от платформы программы
    2. ООП позволяет уменьшить сложность создания ПО
    3. программы, написанные с использованием ООП, выполняются на компьютере быстрее
    4. ООП позволяет увеличить надежность создаваемого ПО
19. Какие утверждения относительно полей класса верны?
    1. final-поле может быть проинициализировано в конструкторе
    2. идентификатор при объявлении становится составным именем поля
    3. числовое поле ссылочного типа имеет значение по умолчанию null
    4. поля не могут иметь модификатор доступа protected
20. Предположим, вы моделируете автомобиль, описывая его свойства в формате Java-класса. Какие из следующих полей нужно объявить динамическими (не static)?
    1. максимально допустимая масса для этого класса автомобилей
    2. дата начала выпуска автомобилей
    3. идентификационный номер (VIN)

4. дата выпуска автомобиля

21. Какие утверждения относительно класса BufferedOutputStream верны?
    1. предоставляет возможность производить многократную запись небольших блоков данных без обращения к устройству вывода
    2. инициирование передачи содержимого буфера на устройство вывода осуществляется самим классом BufferedOutputStream
    3. инициирование передачи содержимого буфера на устройство вывода можно осуществить явным образом
    4. запись с использованием буфера значительно быстрее, чем без него
22. Which of the following can fill in the blank in this code to make it compile? (Отметьте все варианты)

```
public class Ant {
_____ void method() { }
}
```

    1. default
    2. final
    3. private
    4. Public
    5. String
    6. zzz:

23. Что из перечисленных конструкций будет скомпилировано без ошибок?
    1. final static void method4() { }
    2. public final int void method() { }
    3. private void int method() { }
    4. static final void method3() { }
    5. void final method() {}
    6. void public method() { }

24. Какой из перечисленных методов будет успешно компилироваться? (Отметьте все варианты)
    1. public void methodA() { return;}
    2. public void methodB() { return null;}
    3. public void methodD() {}
    4. public int methodD() { return 9;}
    5. public int methodE() { return 9.0;}
    6. public int methodF() { return;}
    7. public int methodG() { return null;}
25. Что из перечисленных конструкций будет скомпилировано без ошибок? (Отметьте все варианты)
    1. public void moreA(int... nums) {}
    2. public void moreB(String values, int... nums) {}
    3. public void moreC(int... nums, String values) {}
    4. public void moreD(String... values, int... nums) {}
    5. public void moreE(String[] values, ...int nums) {}
    6. public void moreF(String... values, int[] nums) {}
    7. public void moreG(String[] values, int[] nums) {}

26. Given the following method, which of the method calls return 2? (Отметьте все варианты)

```
public int howMany(boolean b, boolean... b2) {
return b2.length;
}
```

    1. howMany();
    2. howMany(true);
    3. howMany(true, true);
    4. howMany(true, true, true);
    5. howMany(true, {true});
    6. howMany(true, {true, true});
    7. howMany(true, new boolean[2]);

27. Какие утверждения верны? (Отметьте все варианты)
    1. Package private access is more lenient than protected access.
    2. A public class that has private fields and package private methods is not visible to classes outside the package.
    3. You can use access modifiers so only some of the classes in a package see a particular package private class.
    4. You can use access modifiers to allow read access to all methods, but not any instance variables.
    5. You can use access modifiers to restrict read access to all classes that begin with the word Test.

28. Given the following my.school.ClassRoom and my.city.School class definitions, which line numbers in main() generate a compiler error? (Отметьте все варианты)

```
1:          package my.school;
2:          public class Classroom {
3:           private int roomNumber;
4:          protected String teacherName;
5:          static int globalKey = 54321;
6:          public int floor = 3;
7:          Classroom(int r, String t) {
8:          roomNumber = r;
9:          teacherName = t; } }
1:          package my.city;
2:          import my.school.*;
3:          public class School {
4:          public static void main(String[] args) {
5:          System.out.println(Classroom.globalKey);
6:           Classroom room = new Classroom(101, ""Mrs. Anderson");
7:           System.out.println(room.roomNumber);
8:          System.out.println(room.floor);
9:          System.out.println(room.teacherName); } }
```

1. None, the code compiles fine.
2. Line 5
3. Line 6
4. Line 7
5. Line 8
6. Line 9

29. Какие утверждения верны? (Отметьте все варианты)

1. Encapsulation uses package private instance variables.
2. Encapsulation uses private instance variables.
3. Encapsulation allows setters.
4. Immutability uses package private instance variables.
5. Immutability uses private instance variables.
6. Immutability allows setters.

30. Which are methods using JavaBeans naming conventions for accessors and mutators? (Отметьте все варианты)

1. public boolean getCanSwim() { return canSwim;}
2. public boolean canSwim() { return numberWings;}
3. public int getNumWings() { return numberWings;}
4. public int numWings() { return numberWings;}
5. public void setCanSwim(boolean b) { canSwim = b;}

31. Что выведет (в консоль) следующий код?

```
1: package rope;
2: public class Rope {
3: public static int LENGTH = 5;
4: static {
5: LENGTH = 10;
6: }
7: public static void swing() {
8: System.out.print("swing ");
9: }
10: }
1: import rope.*;
2: import static rope.Rope.*;
3: public class Chimp {
4: public static void main(String[] args) {
5: Rope.swing();
6: new Rope().swing();
7: System.out.println(LENGTH);
8: }
9: }
```

Ответ: swing swing 10

32. Какие утверждения корректны по отношению к данному коду?(Отметьте все варианты)

```
1: public class Rope {
2: public static void swing() {
3: System.out.print("swing ");
4: }
5: public void climb() {
6: System.out.println("climb ");
7: }
8: public static void play() {
9: swing();
10: climb();
11: }
12: public static void main(String[] args) {
13: Rope rope = new Rope();
14: rope.play();
15: Rope rope2 = null;
16: rope2.play();
17: }
18: }
```

1. The code compiles as is.
2. There is exactly one compiler error in the code.
3. If the lines with compiler errors are removed, the output is climb climb.
4. If the lines with compiler errors are removed, the output is swing swing.

33. What modifiers are implicitly applied to all interface methods? (Отметьте все варианты)

1. protected
2. public
3. static
4. void
5. abstract
6. default

34. Что выведет (в консоль) следующий код?

```
1: class Mammal {
2: public Mammal(int age) {
3: System.out.print("Mammal");
4: }
5: }
6: public class Platypus extends Mammal {
7: public Platypus() {
8: System.out.print("Platypus");
9: }
10: public static void main(String[] args) {
11: new Mammal(5);
12: }
13: }
```

1. Platypus
2. Mammal
3. PlatypusMammal
4. MammalPlatypus
5. Compile error in line 8.
6. Compile error in line 11.

35. Что можно вставить на место подчеркивания, чтобы код успешно компилировался? (Отметьте все варианты)

```
public interface CanHop {}
public class Frog implements CanHop {
public static void main(String[] args) {
frog = new TurtleFrog();
}
}
public class BrazilianHornedFrog extends Frog {}
public class TurtleFrog extends Frog {}
```

1. Frog
2. TurtleFrog
3. BrazilianHornedFrog
4. CanHop
5. Object
6. Long

36. Which statement(s) are correct about the following code? (Отметьте все варианты)

```
public class Rodent {
protected static Integer chew() throws Exception {
System.out.println("Rodent is chewing");
return 1;
}
}
public class Beaver extends Rodent {
public Number chew() throws RuntimeException {
System.out.println("Beaver is chewing on wood");
return 2;
}
}
```

1. It will compile without issue.
2. It fails to compile because the type of the exception the method throws is a subclass of the type of exception the parent method throws.
3. It fails to compile because the return types are not covariant.
4. It fails to compile because the method is protected in the parent class and public in the subclass.
5. It fails to compile because of a static modifier mismatch between the two methods.

37. Which of the following may only be hidden and not overridden? (Отметьте все варианты)

1. private instance methods
2. protected instance methods
3. public instance methods
4. static methods
5. public variables
6. private variables

38. Выберите корректное утверждение о данном коде:

```
1: interface HasExoskeleton {
2: abstract int getNumberOfSections();
3: }
4: abstract class Insect implements HasExoskeleton {
5: abstract int getNumberOfLegs();
6: }
7: public class Beetle extends Insect {
8: int getNumberOfLegs() { return 6; }
9: }
```

1. It compiles and runs without issue.
2. Compile error in line 2.
3. Compile error in line 4.
4. Compile error in line 7.
5. It compiles but throws an exception at runtime.

39. Which of the following statements about polymorphism are true? (Отметьте все варианты)

1. A reference to an object may be cast to a subclass of the object without an explicit cast.
2. If a method takes a superclass of three objects, then any of those classes may be passed as a parameter to the method.
3. A method that takes a parameter with type java.lang.Object will take any reference.
4. All cast exceptions can be detected at compile-time.
5. By defining a public instance method in the superclass, you guarantee that the specific method will be called in the parent class at runtime.

40. Which statements are true for both abstract classes and interfaces? (Отметьте все варианты)

1. All methods within them are assumed to be abstract.
2. Both can contain public static final variables.
3. Both can be extended using the extend keyword.

4. Both can contain default methods.
5. Both can contain static methods.
6. Neither can be instantiated directly.
7. Both inherit java.lang.Object.

Правильные ответы к заданию 1.1

1	2	3	4	5	6	7	8	9	10
б	бг	в	б	абвд	а	б	а	а	в
11	12	13	14	15	16	17	18	19	20
в	вг	г	б	в	вг	бвг	бг	ав	вг
21	22	23	24	25	26	27	28	29	30
аг	в	аг	вг	абж	гж	г	бвге	бвд	вд
31	32	33	34	35	36	37	38	39	40
	бг	б	д	абгд	вд	агде	г	бв	бвде

Ответ к заданию 31: swing swing 10.

Оценочный лист к заданию 1.1.

Критерий	Максимальное количество баллов
Вопрос 1-40	1

Задание 1.2.
Содержание задания:

Создайте программу, осуществляющую чтение текстового файла и загрузку всех его слов (разделители – пробелы, знаки препинания удаляются) в соответствующую коллекцию. Организовать подсчет чаще всего встречающегося слова. Интерфейс программы – нативное десктоп-приложение.

В программе должны присутствовать методы для чтения и загрузки слов файла в коллекцию, для преобразования коллекции в структуру «ключ»-«значение», где «значение» - частота встречаемости слова, для поиска чаще всего встречающихся слов. Предусмотреть автоматизированное тестирование средствами JUnit всех указанных методов.

Окно диалога программы должно включать поле ввода для имени файла (с вызовом диалога открытия файла ОС), а также текстовое поле для вывода чаще всего встречающихся слов. Провести ручное тестирование для случаев а) корректного и некорректного имени файла б) пустого файла в) файла, в которых слова встречаются по одному разу.

Оформить результаты эргономического тестирования приложения.

Оценочный лист к заданию 1.2.

Показатель результативности	Индикатор ПК-1.2	Максимальное количество баллов
Приложение создано, отлажено, работает, проходит тесты при корректных входных данных	ПК-1.2	5
Подготовлены тесты JUnit и успешно пройдены	ПК-1.2	5
Выполнено (и пройдено) ручное тестирование	ПК-1.2	5
Дано развернутое методическое объяснение применению концепций и возможностей ООП	ПК-1.2	5

Методические материалы, определяющие процедуру и критерии оценивания сформированности компетенций при проведении промежуточной аттестации

Код контролируемой компетенции (индикаторы)	Наименование оценочного средства	Максимальное количество баллов	Всего баллов	Уровень освоения компетенции (в баллах)		
				Пороговый (56-70%)	Продвинутый (71-85%)	Высокий (86-100%)
ПК-1.1	Задание А 1-40	40	40	20-24	28-32	36-40
ПК-1.2	Задание В	20	20	10-14	15-16	17-20

Полученное число баллов (30-60) выставляется в графу «Промежуточная аттестация» балльно-рейтинговой карты дисциплины.