



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО  
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

**(Институт технологий (филиал) ДГТУ в г. Волгодонске)**



**УТВЕРЖДАЮ**

И.о. директора

Н.М. Сидоркина

«22» апреля 2024 г.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
(ОЦЕНОЧНЫЕ СРЕДСТВА)**  
для проведения текущего контроля и промежуточной аттестации  
по дисциплине  
«Межплатформенное программирование»  
для обучающихся по направлению подготовки  
*09.03.02 Информационные системы и технологии*  
программа бакалавриата «Информационные системы»  
2024 год набора

Волгодонск  
2024

## Лист согласования

Оценочные материалы (оценочные средства) по дисциплине \_\_\_\_\_ Межплатформенное программирование  
(наименование)

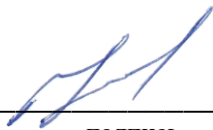
составлены в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки (специальности)

\_\_\_\_\_ 09.03.02 Информационные системы и технологии \_\_\_\_\_,  
(код направления (специальности), наименование)


Рассмотрены и одобрены на заседании кафедры «ТСиИТ» протокол № 9 от «22» апреля 2024 г.

Разработчики оценочных материалов (оценочных средств)

Доцент

  
\_\_\_\_\_ Н.В. Кочковая  
подпись

Заведующий кафедрой

  
\_\_\_\_\_ Н.В. Кочковая  
подпись

—  
**Согласовано:**  
Директор НПЦ  
«Микроэлектроника»

  
\_\_\_\_\_ С.Л. Бондаренко  
подпись

Начальник отдела ПО  
ООО «Топаз-сервис»

  
\_\_\_\_\_ Д.В. Чубукин  
подпись

## 1 Содержание

1 Паспорт оценочных материалов (оценочных средств)	4
1.1 Перечень компетенций, формируемых дисциплиной (модулем), с указанием этапов их формирования в процессе освоения ОПОП	4
1.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования	9
1.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, описание шкал оценивания	9
2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы	12
2.1 Задания для оценивания результатов обучения в виде знаний	12
2.2 Задания для оценивания результатов в виде владений и умений	15
2.3 Типовые экзаменационные материалы	20

## **1 Паспорт оценочных материалов (оценочных средств)**

Оценочные материалы (оценочные средства) прилагаются к рабочей программе дисциплины и представляет собой совокупность контрольно-измерительных материалов (типовые задачи (задания), контрольные работы, тесты и др.) и методов их использования, предназначенных для измерения уровня достижения обучающимися установленных результатов обучения.

Оценочные материалы (оценочные средства) используются при проведении текущего контроля успеваемости и промежуточной аттестации обучающихся.

### **1.1 Перечень компетенций, формируемых дисциплиной, с указанием этапов их формирования в процессе освоения ОПОП**

Перечень компетенций, формируемых в процессе изучения дисциплины:

ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение.

ПК-6: Способность проектировать ИС, в том числе по профилю подготовки.

Конечными результатами освоения дисциплины являются сформированные когнитивные дескрипторы «знать», «уметь», «владеть», расписанные по отдельным компетенциям. Формирование дескрипторов происходит в течение всего семестра по этапам в рамках контактной работы, включающей различные виды занятий и самостоятельной работы, с применением различных форм и методов обучения (табл. 1).

Таблица 1 – Формирование компетенций в процессе изучения дисциплины

Код компетенции	Уровень	Дескрипторы компетенции (результаты обучения, показатели достижения результата обучения, которые обучающийся может продемонстрировать)	Вид учебных занятий, работы <sup>1</sup> , формы и методы обучения, способствующие формированию и развитию компетенции <sup>2</sup>	Контролируемые разделы и темы дисциплины <sup>3</sup>	Оценочные материалы (оценочные средства), используемые для оценки уровня сформированности компетенции
ПК-5	Знает технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах; основы объектно-ориентированного подхода к программированию	принципы моделирования, классификацию способов представления моделей систем; приемы, методы, способы формализации объектов, процессов, явлений и реализацию их на компьютере; достоинства и недостатки различных способов представления моделей систем; разработку алгоритмов фиксации и обработки результатов моделирования систем; способы планирования машинных экспериментов с моделями	Лек, Пр., Ср лекция	1.1, 1.2, 1.3, 1.4., 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10	Контрольные вопросы
	Умеет работать с современными системами программирования, включая объектно-ориентированные; ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования	использовать технологии моделирования; представлять модель в математическом и алгоритмическом виде; оценивать качество модели; показывать теоретические основания модели;	Лек, Пр., Ср Еработа в малых группах, анализ практических работ		Практическая работа

	Владеет языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки и тестирования программ не менее, чем на одном из алгоритмических процедурных языков программирования высокого уровня	Построением имитационных моделей информационных процессов; получением концептуальных моделей систем; построением моделирующих алгоритмов;	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа
ПК-6	Знает основные методы проектирования ИС, профили открытых ИС, функциональные и технологические стандарты разработки ИС, виды проектных решений; основные потребности и требования к ИС организаций по профилю подготовки, исходя из действующих правовых норм	Методы, модели и современные инструментальные средства исследования для оценки и обеспечения надежности и качества информационных систем, основы разработки средств обнаружения, локализации, и восстановления отказавших элементов информационных систем	Лек, Пр., Ср лекция	1.1, 1.2, 1.3, 1.4., 1.5, 1.6, 1.7, 1.8, 1.9, 1.10, 1.11, 1.12, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 2.10	Контрольные вопросы
	Умеет проектировать объекты профессиональной деятельности с применением основных базовых и информационных технологий; проектировать ИС и подсистемы по профилю подготовки, исходя из действующих правовых норм	Разрабатывать и исследовать по критериям надежности избыточные информационные структуры, разрабатывать математические модели надежности информационных систем, разрабатывать средства обнаружения, локализации и восстановления отказавших элементов информационных систем	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа

Владеет навыками применения проектных решений ИС	Инструментальными средствами обработки информации	Лек, Пр., Ср работа в малых группах, анализ практических работ		Практическая работа
--	---	---	--	---------------------

## **1.2 Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

Оценивание результатов обучения по дисциплине осуществляется в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся.

По дисциплине «Межплатформенное программирование» предусмотрены следующие виды контроля: текущий контроль (осуществление контроля всех видов аудиторной и внеаудиторной деятельности обучающегося с целью получения первичной информации о ходе усвоения отдельных элементов содержания дисциплины); промежуточная аттестация (оценивается уровень и качество подготовки по дисциплине в целом).

## **1.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы обучающихся. Текущий контроль служит для оценки объёма и уровня усвоения обучающимся учебного материала одного или нескольких разделов дисциплины (модуля) в соответствии с её рабочей программой и определяется результатами текущего контроля знаний обучающихся.

Текущий контроль осуществляется два раза в семестр по календарному графику учебного процесса.

Текущий контроль предполагает начисление баллов за выполнение различных видов работ. Результаты текущего контроля подводятся по шкале балльно-рейтинговой системы. Регламент балльно-рейтинговой системы определен Положением о системе «Контроль успеваемости и рейтинг обучающихся».

Текущий контроль является результатом оценки знаний, умений, навыков и приобретенных компетенций обучающихся по всему объёму учебной дисциплины, изученному в семестре, в котором стоит форма контроля в соответствии с учебным планом.

Текущий контроль успеваемости предусматривает оценивание хода освоения дисциплины: теоретических основ и практической части.

Промежуточная аттестация по дисциплине «Межплатформенное программирование» проводится в форме экзамена.

В табл. 2 приведено весовое распределение баллов и шкала оценивания по видам контрольных мероприятий.



Таблица 2 – Весовое распределение баллов и шкала оценивания по видам контрольных мероприятий для экзамена

Текущий контроль (50 баллов <sup>1</sup> )				Промежуточная аттестация (50 баллов)	Итоговое количество баллов по результатам текущего контроля и промежуточной аттестации
Блок 1		Блок 2			
Лекционные занятия (X <sub>1</sub> )	Практические занятия (Y <sub>1</sub> )	Лекционные занятия (X <sub>2</sub> )	Практические занятия (Y <sub>2</sub> )	от 0 до 50 баллов	Менее 41 балла – неудовлетворительно; 41-60 баллов – удовлетворительно; 61-80 баллов – хорошо; 81-100 баллов – отлично
5	15	5	25		
Сумма баллов за 1 блок = X <sub>1</sub> + Y <sub>1</sub> = 20		Сумма баллов за 2 блок = X <sub>2</sub> + Y <sub>2</sub> = 30			

Для определения фактических оценок каждого показателя выставляются следующие баллы (табл.3):

Таблица 3– Распределение баллов по дисциплине

Вид учебных работ по дисциплине	Количество баллов	
	1 блок	2 блок
<i>Текущий контроль (50 баллов)</i>		
Выполнение практических работ в виде проведения исследования, оформления отчета	<b>20</b>	<b>20</b>
Защита практических работ в форме собеседования по контрольным вопросам	<b>5</b>	<b>5</b>
<i>Промежуточная аттестация (50 баллов)</i>		
<p>Экзамен по дисциплине «Межплатформенное программирование» проводится в письменной форме в виде ответов на вопросы для промежуточной аттестации. Задание для зачета состоит из 3 вопросов. Первый и второй вопрос позволяют проконтролировать знания обучающегося, третий – умения и навыки. Правильные ответы на первый и второй вопросы оцениваются в 15 баллов, третий – в 20 баллов. За неверно выполненное задание – 0 баллов.</p>		

<sup>1</sup> Вид занятий по дисциплине (лекционные, практические, лабораторные) определяется учебным планом. Количество столбцов таблицы корректируется в зависимости от видов занятий, предусмотренных учебным планом.

Распределение баллов по блокам, по каждому виду занятий в рамках дисциплины определяет преподаватель. Распределение баллов по дисциплине утверждается протоколом заседания кафедры. По заочной форме обучения мероприятия текущего контроля не предусмотрены.

<b>Сумма баллов по дисциплине 100 баллов</b>
--

Экзамен является формой итоговой оценки качества освоения обучающимся образовательной программы по дисциплине в целом или по разделу дисциплины. По результатам экзамена обучающемуся выставляется оценка «отлично», «хорошо», «удовлетворительно», или «неудовлетворительно».

Оценка «отлично» (81-100 баллов) выставляется обучающемуся, если:

- обучающийся набрал по текущему контролю необходимые и достаточные баллы для выставления оценки автоматом ;
- обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения;
- обучающийся анализирует элементы, устанавливает связи между ними, сводит их в единую систему, способен выдвинуть идею, спроектировать и презентовать свой проект (решение);
- ответ обучающегося по теоретическому и практическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, и удовлетворяет требованиям программы дисциплины;
- обучающийся продемонстрировал свободное владение концептуально-понятийным аппаратом, научным языком и терминологией соответствующей дисциплины;
- на дополнительные вопросы преподавателя обучающийся дал правильные ответы.

Компетенция (и) или ее часть (и) сформированы на высоком уровне (уровень 3) (см. табл. 1).

Оценка «хорошо» (61-80 баллов) выставляется обучающемуся, если:

- обучающийся знает, понимает основные положения дисциплины, демонстрирует умение применять их для выполнения задания, в котором нет явно указанных способов решения; анализирует элементы, устанавливает связи между ними;
- ответ по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, является полным, или частично полным и удовлетворяет требованиям программы, но не всегда дается точное, уверенное и аргументированное изложение материала;
- на дополнительные вопросы преподавателя обучающийся дал правильные ответы;
- обучающийся продемонстрировал владение терминологией соответствующей дисциплины.

Компетенция (и) или ее часть (и) сформированы на среднем уровне (уровень 2) (см. табл. 1).

Оценка «удовлетворительно» (41-60 баллов) выставляется обучающемуся, если:

- обучающийся знает и воспроизводит основные положения дисциплины в соответствии с заданием, применяет их для выполнения типового задания в котором очевиден способ решения;

- обучающийся продемонстрировал базовые знания важнейших разделов дисциплины и содержания лекционного курса;

- у обучающегося имеются затруднения в использовании научно-понятийного аппарата в терминологии курса;

- несмотря на недостаточность знаний, обучающийся имеется стремление логически четко построить ответ, что свидетельствует о возможности последующего обучения.

Компетенция (и) или ее часть (и) сформированы на базовом уровне (уровень 1) (см. табл. 1).

Оценка «неудовлетворительно» (менее 41 балла) выставляется обучающемуся, если:

- обучающийся имеет представление о содержании дисциплины, но не знает основные положения (темы, раздела, закона и т.д.), к которому относится задание, не способен выполнить задание с очевидным решением, не владеет навыками применения средств проектирования информационных систем и технологий.

- у обучающегося имеются существенные пробелы в знании основного материала по дисциплине;

- в процессе ответа по теоретическому материалу, содержащемуся в вопросах экзаменационного билета, допущены принципиальные ошибки при изложении материала.

Компетенция(и) или ее часть (и) не сформированы.

## **2 Контрольные задания (демоверсии) для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы**

### **2.1 Задания для оценивания результатов обучения в виде знаний**

Контроль знаний по дисциплине «Межплатформенное программирование» осуществляется посредством тестовых заданий.

База тестовых вопросов по материалу курса

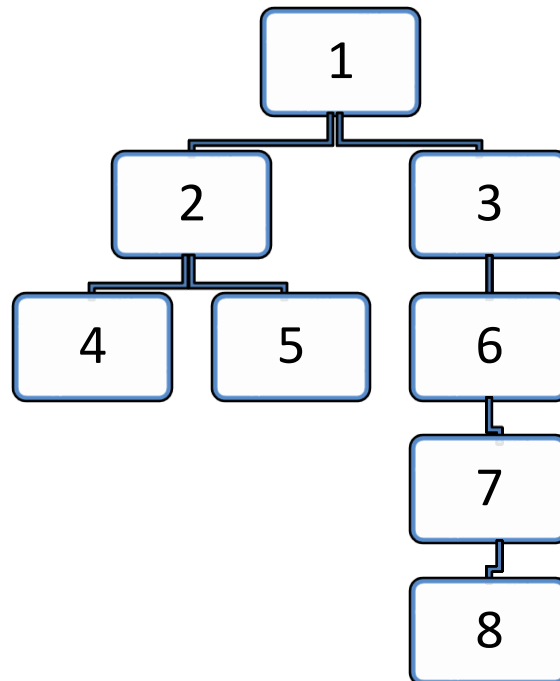
1. Для неупорядоченного множества элементов применяется следующий метод поиска:
  - a. «Пополамного» деления (метод дихотомии)
  - b. Последовательного перебора
  - c. Поиск с возвратом
2. Метод сортировки «пузырек» для множества из  $N$  элементов имеет в худшем случае следующую вычислительную сложность:
  - a.  $\log N$
  - b.  $N$
  - c.  $N^2$

- d.  $2^N$
- 3. Наиболее затратной с точки зрения вычислительной сложности является следующая асимптотическая зависимость:
  - a.  $N!$
  - b.  $2^N$
  - c.  $\log N$
- 4. Объем оперативной памяти, необходимой для программной реализации алгоритма, характеризует:
  - a. Временную сложность алгоритма
  - b. Пространственную сложность алгоритма
  - c. Вычислительную сложность алгоритма
- 5. Выбор наилучшего с какой-то точки зрения варианта на каждом шаге характерен для:
  - a. Алгоритмов динамического программирования
  - b. Алгоритмов «разделяй и властвуй»
  - c. «Жадных» алгоритмов
- 6. Абстрактная структура данных, в которой данные помещаются и удаляются из одного места, называется:
  - a. Списком
  - b. Деревом
  - c. Стеком
- 7. Выбор необходимого из множества возможных вариантов на каждом шаге характерен для:
  - a. Алгоритмов динамического программирования
  - b. Алгоритмов «разделяй и властвуй»
  - c. «Жадных» алгоритмов
- 8. Упорядочивание списка элементов некоторого множества называется:
  - a. Фильтрацией
  - b. Сортировкой
  - c. Выборкой
- 9. Рекурсивные процедуры реализуются на основе абстрактного типа данных:
  - a. Очередь
  - b. Приоритетная очередь
  - c. Стек
- 10. Функция, определенная на множестве элементов одного типа, и принимающая значения из множества элементов другого типа, является абстрактным типом данных:
  - a. Наследование

б. Отображение

с. Очередь

11. Для дерева, показанного на рисунке,



высота дерева равна:

а. 3

б. 4

с. 8

д. 5

12. Обход дерева, при котором вначале посещается корень дерева, а затем последовательно все узлы для каждого из поддеревьев, называется:

а. Прямым обходом

б. Симметричным обходом

с. Обратным обходом

13. Позволяет ли «жадный» алгоритм найти оптимальное решение:

а. Только «рациональное»

б. Позволяет всегда

с. Позволяет не всегда

14. Если перед работой алгоритма поиска выполняется сортировка данных, то это применение метода:

а. Эвристик

б. Динамического программирования

с. «Разделяй и властвуй»

15. Если в алгоритме последовательно для  $N$  элементов выполняется одно циклическое вычисление, а затем другое, то вычислительная сложность асимптотически равна:
- $N*N$
  - $N+N$
  - $2^N$

## 2.2 Задания для оценивания результатов в виде владений и умений

Контроль умений и навыков по дисциплине «Межплатформенное программирование» осуществляется посредством подготовки выполнения практических работ и экзамена.

Примерная тематика практических работ:

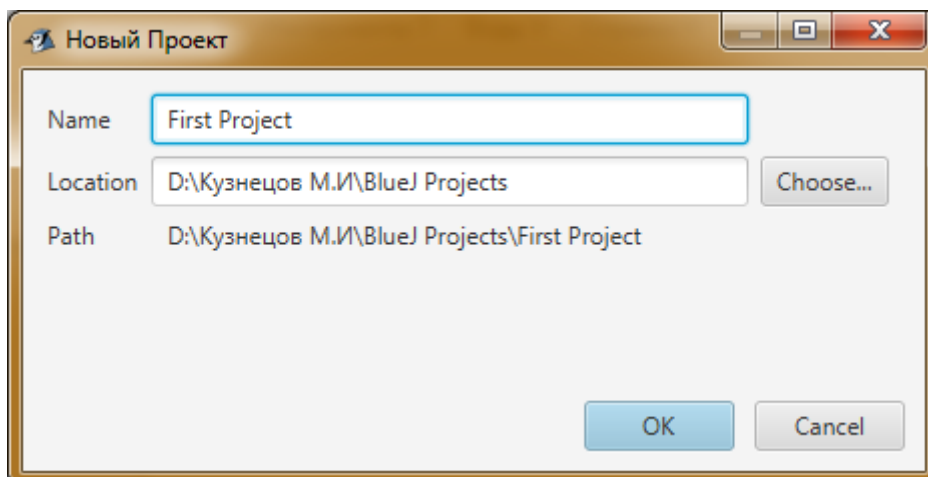
1. Создание Java приложения
2. Основные типы и операторы языка Java
3. Работа с классами
4. Работа с файлами
5. Приложение "клиент - сервер"
6. Распараллеливание и многопоточность
7. Работа со строками в Java /
8. Работа с коллекциями в Java
9. Создание графического интерфейса

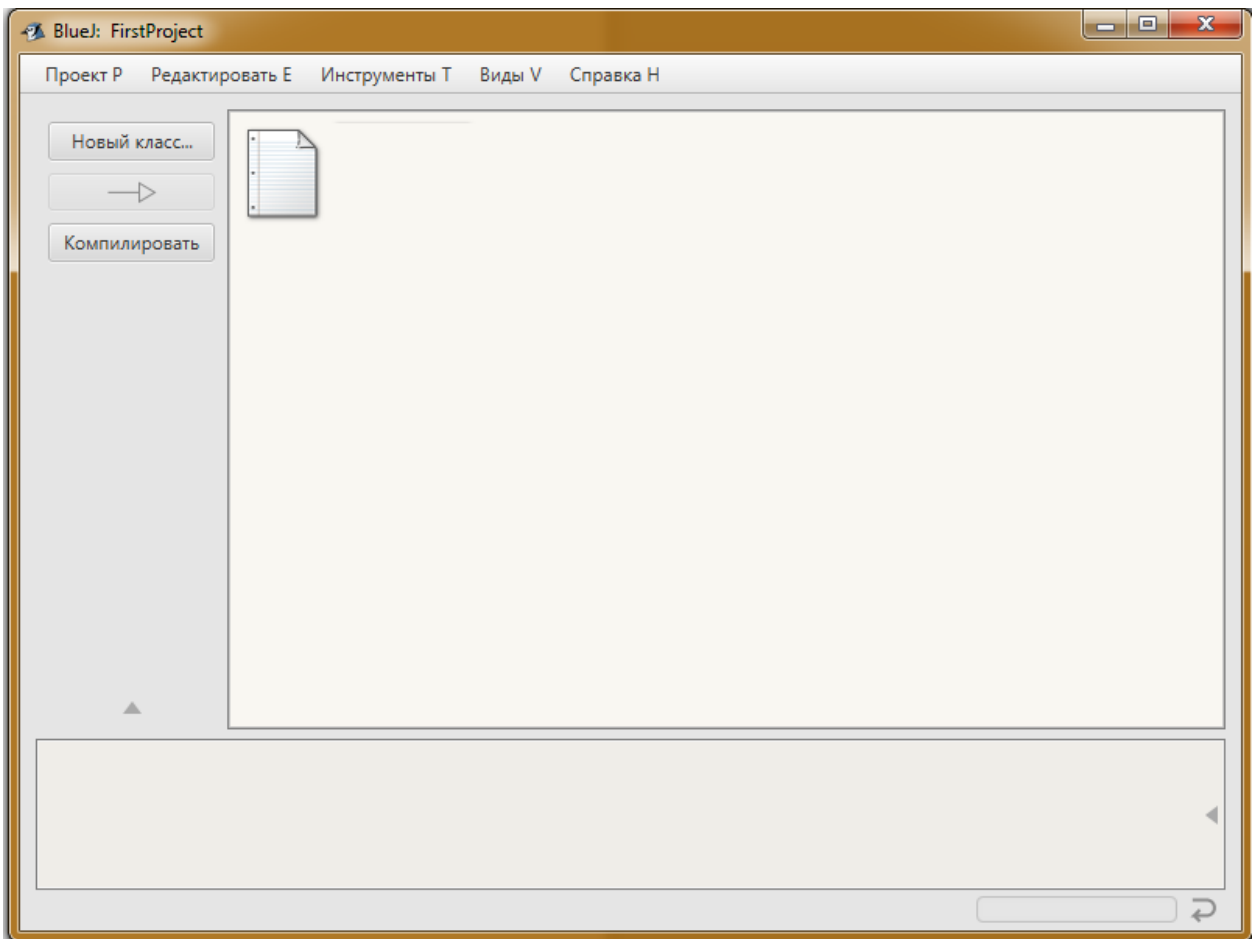
Типовой пример практического задания по дисциплине «Межплатформенное программирование».

## Знакомство с BlueJ.

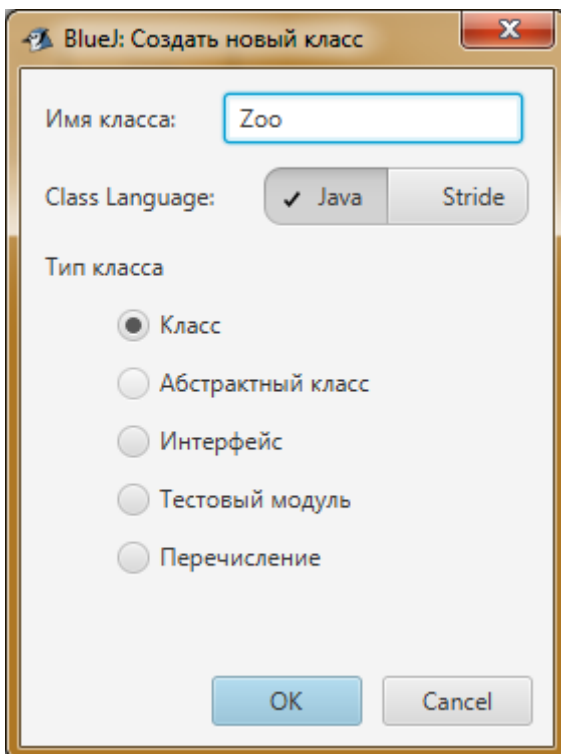
### Первый вводный проект.

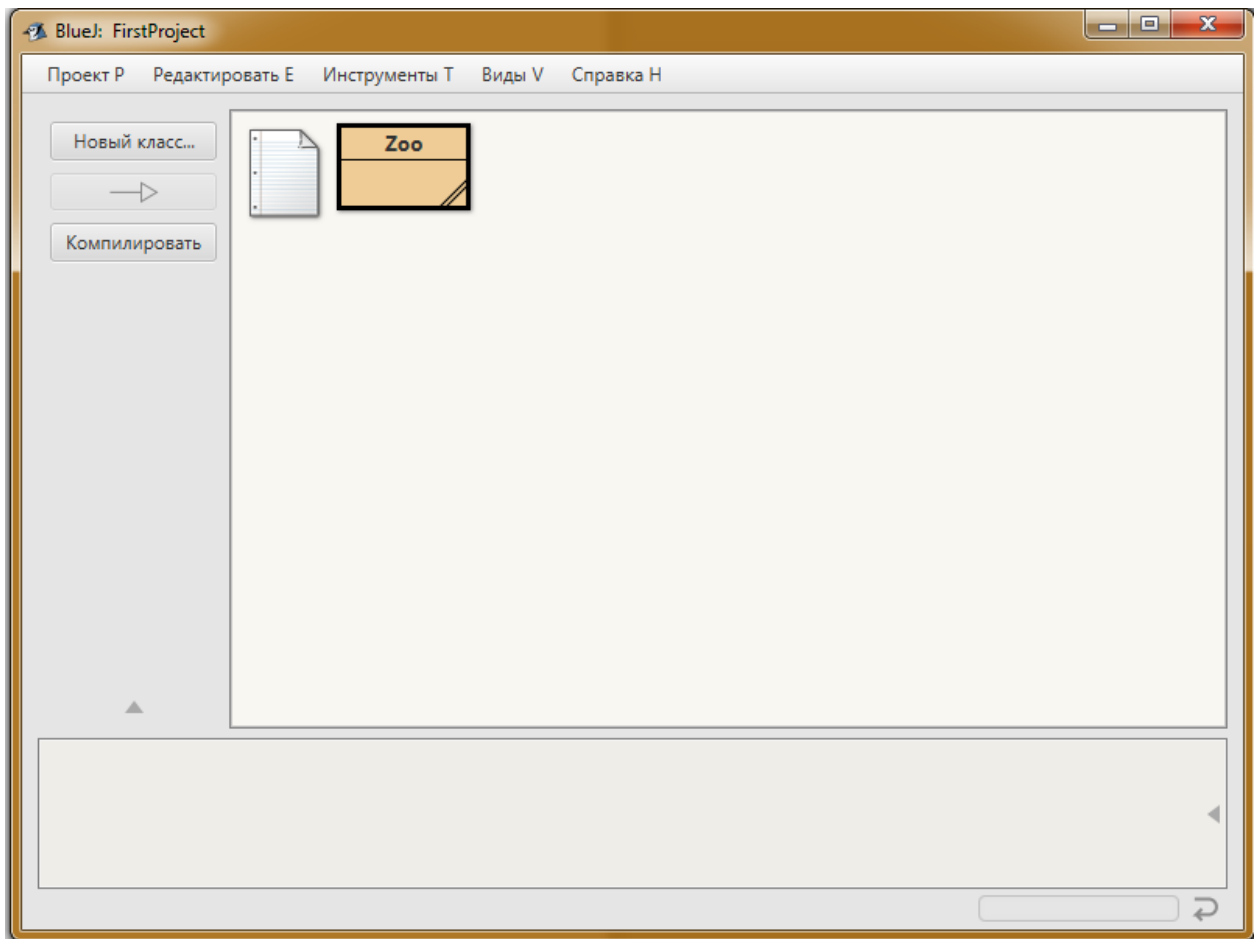
1. Для начала создаем новый проект.





2. Создаем новый класс и вводим в него наш код-пример:





3. После написания кода, снизу выводится сообщение «Класс скомпилирован – синтаксических ошибок нет».



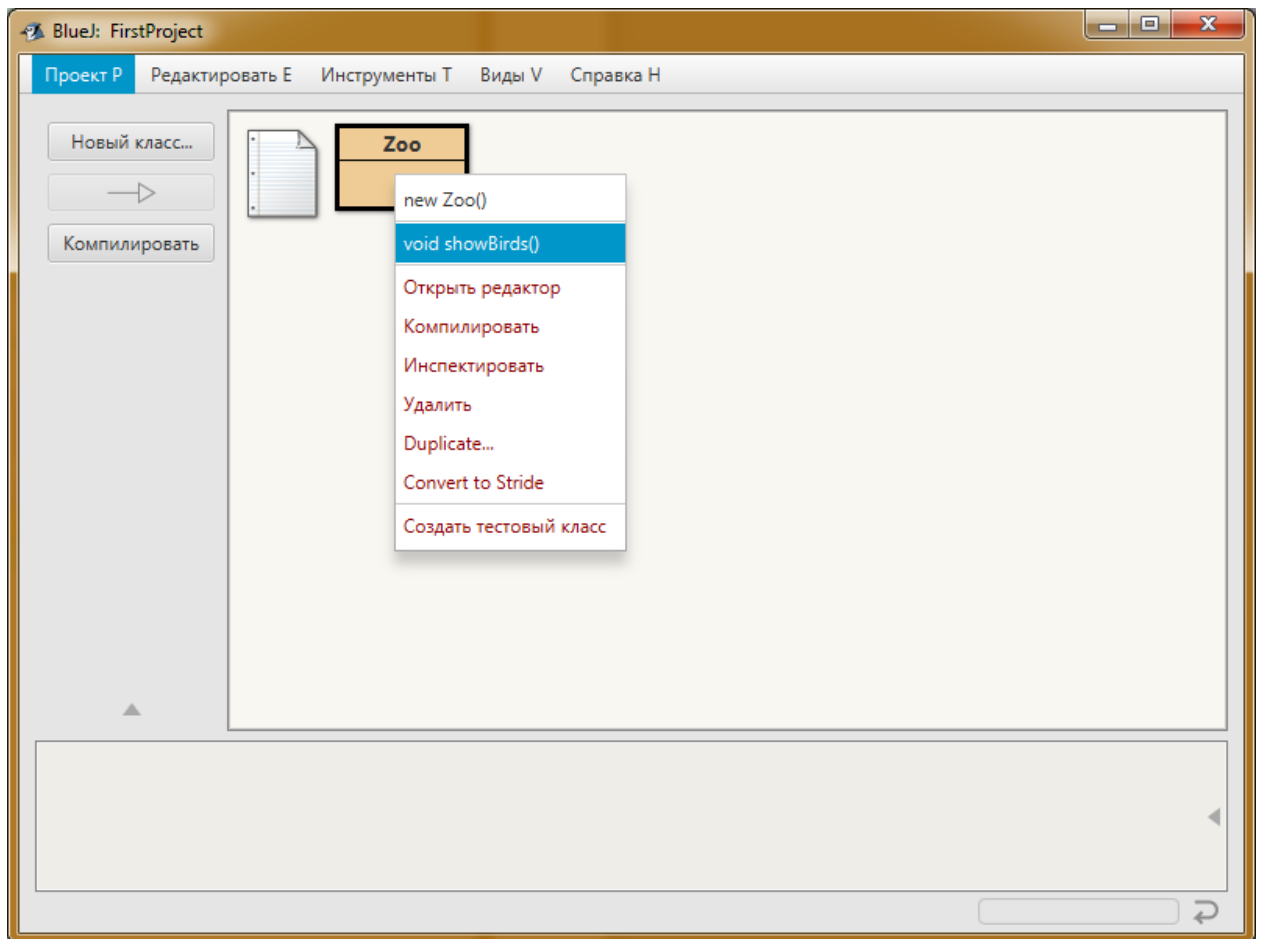
```
/**
 * Show all the birds.
 *
 * @author Maxud
 * @version 0.01
 */
public class Zoo
{
    public static void showBirds()
    {
        String[] birds = new String[]{"Гусь", "Селезень", "Утка"};

        System.out.println("У нас есть " + birds.length + " птицы: ");

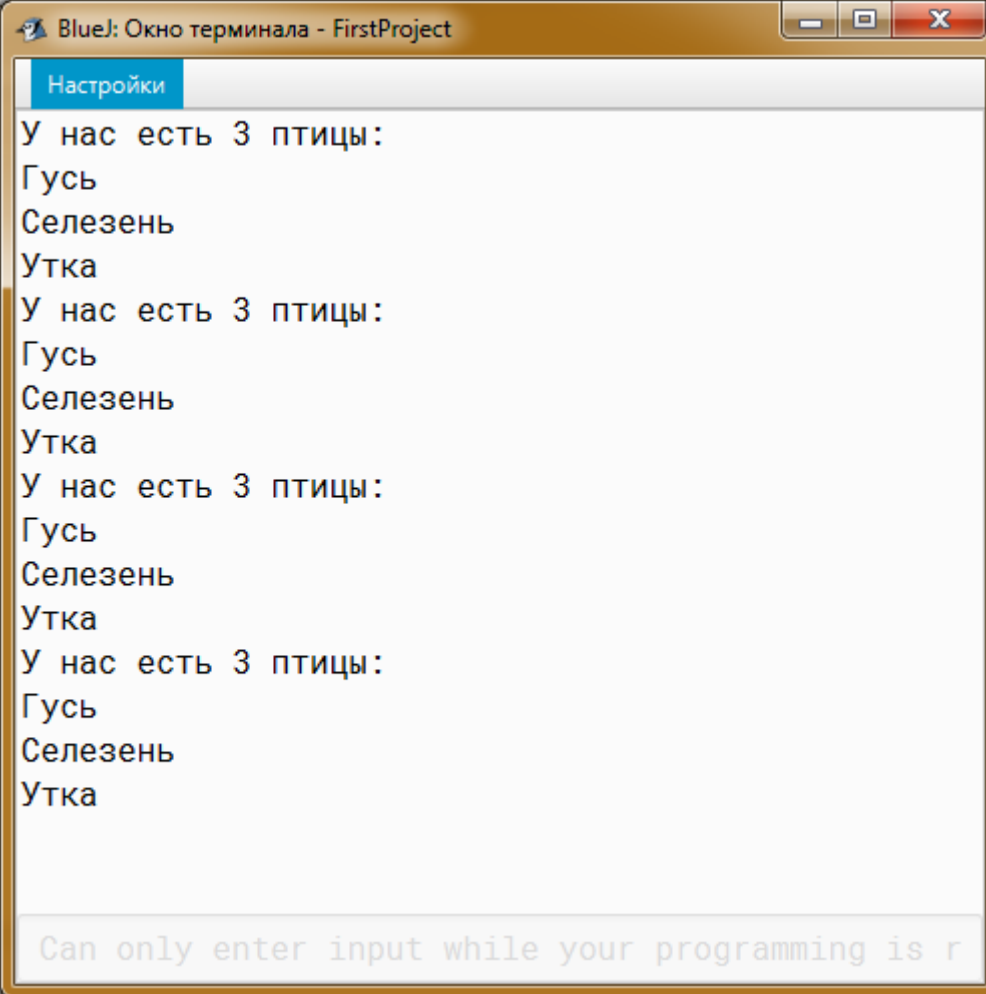
        for(int i = 0; i < birds.length; i++)
        {
            System.out.println(birds[i]);
        }
    }
}
```

Класс откомпилирован - синтаксических ошибок нет | Сохранено

4. Закрываем окно с кодом и в главном окне проекта, нажимаем на наш класс “Zoo” правой кнопкой мыши и выбираем void “showBirds()”.



5. Если мы все верно написали, то нам выведется наш список птиц и их количество в окне терминала.



```
BlueJ: Окно терминала - FirstProject
Настройки
У нас есть 3 птицы:
Гусь
Селезень
Утка
У нас есть 3 птицы:
Гусь
Селезень
Утка
У нас есть 3 птицы:
Гусь
Селезень
Утка
У нас есть 3 птицы:
Гусь
Селезень
Утка
Can only enter input while your programming is r
```

## 2.3 Типовые экзаменационные материалы

Перечень примерных вопросов к экзамену

1. История создания кроссплатформенных систем программирования.
2. Основные направления Java.
3. JDK и JRE.
4. Среды разработки для Net.
5. Синтаксис языка C#.
6. Типы данных в Java, C#, C++, Python.
7. Абстракции и объекты.
8. Абстрактные классы.
9. Интерфейсы.
10. Управление доступом.
11. Инкапуляция.
12. Наследование и полиморфизм.
13. Коллекции объектов.
14. Обработка ошибок и исключения.
15. Внутренние и анонимные (безымянные) внутренние классы.
16. Система ввода-вывода.
17. События и их обработка

18. Библиотека Swing.
19. Концепция Model-View-Controller.
20. Диспетчеры компоновки.
21. Библиотека Qt.
22. Библиотека wxWidgets.
23. Работа с сетевыми протоколами.
24. Интернационализация.
25. Работа с базами данных.
26. Работа со звуком и графикой.
27. Удаленный вызов методов.

Полнота ответа на поставленный вопрос, умение использовать термины, формулы, приводить примеры, делать выводы и анализировать конкретные ситуации.

Шкала оценивания

Максимальное количество баллов, которое обучающийся может получить за промежуточную аттестацию (зачет) составляет 100 баллов.

- Менее 41 балла –  
неудовлетворительно;
- 41-60 баллов –  
удовлетворительно;
- 61-80 баллов –  
хорошо;
- 81-100 баллов –  
отлично

**Задачи для экзамена по курсу «Высокоуровневые методы информатики и программирования»**

**Утверждено**

**Протокол № 1 от 28.08.17**

Примеры тестовых задач:

1. В какой кодировке под символ отводится 2 байта?

В кодировке ASCII.

В кодировке UNICODE.

2. Пусть значения целочисленных переменных  $x$  и  $y$  равны 1 и 2 соответственно. Указать значение логического выражения  $(x \geq 1 \text{ и } y < 0)$  или  $(x \leq 1 \text{ и } y > 0)$

Ложь.

Истина.

3. Какой из перечисленных подходов к программированию является наиболее поздним (зрелым)?

структурный

объектно-ориентированный

компонентно-ориентированный

4. Что отличает императивные языки программирования от декларативных?

степень зависимости от среды реализации

стиль программирования

структура программы

5. В чем состоит особенность языков объектно-ориентированного программирования?

этот класс языков основан на сценариях

этот класс языков концептуально близок к любой предметной области

этот класс языков является наиболее машинно-независимым

6. Что понимается под термином "наследование"?

свойство производного объекта сохранять атрибуты базового

свойство производного объекта сохранять операции базового

свойство производного объекта сохранять поведение базового

7. Свойства какого объекта наследует подкласс с неявным базовым классом?

свойства класса object

свойства структуры

свойства интерфейса

3. Сколько раз будет выполнено тело цикла в приведенной ниже программе? Многоточием обозначен фрагмент, не содержащий переменной x.

x := 0;

цикл пока  $x < 1000$

```
| ...  
|  $x := x + 1$ ;  
конец цикла
```

Тело цикла будет выполнено 999 раз.

Тело цикла будет выполнено 1000 раз.

Тело цикла будет выполнено 1001 раз.

8. Пусть  $A = A(x)$  — некоторое условие, зависящее только от значения переменной  $x$ . Указать, чему может быть равно значение переменной  $y$  в результате выполнения следующего фрагмента программы:

```
 $x := 1$ ;  
 $y := 1$ ;  
цикл пока  $A(x)$   
| ...  
| если  $y < 0$   
| | то  
| |  $x := 2$ ;  
| |  $y := 10$ ;  
| | иначе  
| |  $x := 1$ ;  
| |  $y := 20$ ;  
| конец если  
конец цикла
```

Значение  $y$  равно 1 или 10.

Значение  $y$  равно 1 или 20.

Значение  $y$  может быть равным любому из чисел 1, 10, 20.

9. Чему равно значение целочисленной переменной  $x$  в результате выполнения приведенного ниже фрагмента программы?

```
 $x := 64$ ;  
цикл пока  $x * x > 100$   
|  $x := -(x / 2)$ ;  
конец цикла
```

Значение  $x = 16$ .

Значение  $x = 8$ .

Значение  $x = -8$ .

Значение  $x = 4$ .

10. Завершится ли когда-нибудь выполнение цикла в приведенном ниже фрагменте программы?

$x := 1$ ;

цикл пока  $x \neq 144$

|  $x := x * 13$ ;

| если  $x \leq 299$

| | то  $x := x - 299$ ;

| конец если

конец цикла

Завершится.

Не завершится.

Структура экзаменационного билета

Экзаменационный билет включает в себя 2 теоретических вопроса и 1 практическое задание.



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ  
ФЕДЕРАЦИИ

**ИНСТИТУТ ТЕХНОЛОГИЙ (ФИЛИАЛ) ФЕДЕРАЛЬНОГО  
ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
В Г. ВОЛГОДОНСКЕ РОСТОВСКОЙ ОБЛАСТИ**

**(Институт технологий (филиал) ДГТУ в г. Волгодонске)**

Факультет «Технологии и менеджмент»

Кафедра «Технический сервис и информационные технологии»

**ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1**

на 2022/ 2023 учебный год

Дисциплина Межплатформенное

программирование

1. Типы данных в Java, C#, C++, Python \_\_\_\_\_
2. Библиотека Swing \_\_\_\_\_
3. Задача \_\_\_\_\_

Зав. кафедрой \_\_\_\_\_ Н.В. Кочковая

подпись

15.12.2022

дата

АКТУАЛЬНО НА

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

20\_\_/20\_\_уч.год

подпись

Ф.И.О. зав. каф.

## Карта тестовых заданий

**Компетенция:** ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение

**Дисциплина:** Межплатформенное программирование

### Описание теста:

1. Тест состоит из 85 заданий, которые проверяют уровень освоения компетенций обучающегося. При тестировании каждому обучающемуся предлагается 30 тестовых заданий по 15 открытого и закрытого типов разных уровней сложности.

2. За правильный ответ тестового задания обучающийся получает 1 условный балл, за неправильный ответ – 0 баллов. По окончании тестирования, система автоматически определяет «заработанный итоговый балл» по тесту, согласно критериям оценки

3 Максимальная общая сумма баллов за все правильные ответы составляет – 100 баллов.

4. Тест успешно пройден, если обучающийся правильно ответил на 70% тестовых заданий (61 балл).

5. На прохождение тестирования, включая организационный момент, обучающимся отводится не более 45 минут. На каждое тестовое задание в среднем по 1,5 минуты.

6. Обучающемуся предоставляется одна попытка для прохождения компьютерного тестирования.

**Кодификатором** теста по дисциплине является раздел рабочей программы «4. Структура и содержание дисциплины (модуля)»

### Комплект тестовых заданий

#### Задания закрытого типа

#### Задания альтернативного выбора

*Выберите **один** правильный ответ*

#### Простые (1 уровень)

1 Объявление переменной в C# имеет следующий синтаксис

A) var x = value;

B) x value;

B) int x = value;

Г) x: value;

2 Оператор "using" в C# предназначен для того чтобы

A) распорядиться объектом и освободить его ресурсы

B) включить пространство имен в текущий файл

B) открыть файл для чтения или записи

Г) создать экземпляр класса

3 Создание класса в C# имеет следующий синтаксис

A) class MyClass

B) function MyClass

B) struct MyClass

Г) object MyClass



- 4 Разница между циклами "while" и "do-while" в C# заключается в том что \_\_\_\_
- А) цикл while проверяет условие перед выполнением цикла, а цикл do-while выполняет цикл до проверки условия
  - Б) цикл while можно использовать только с массивами, а цикл do-while - с любым типом данных
  - В) цикл while быстрее цикла do-while
  - Г) разницы нет, они взаимозаменяемы
- 5 Оператор "is" в C# предназначен для того чтобы
- А) проверить, является ли объект экземпляром определенного класса или интерфейса
  - Б) проверить, равны ли две переменные
  - В) проверить, не является ли переменная нулевой
  - Г) проверить, имеет ли переменная определенный тип данных

### Средне-сложные (2 уровень)

- 6 Создание массива в C# имеет следующий синтаксис
- А) `int[] myArray = new int[5];`
  - Б) `myArray = {1, 2, 3, 4, 5};`
  - В) `array myArray = {1, 2, 3, 4, 5};`
  - Г) `int myArray[5] = {1, 2, 3, 4, 5};`
- 7 Создание исключения в C# имеет следующий синтаксис
- А) `throw new Exception();`
  - Б) `raise Exception();`
  - В) `throw Exception();`
  - Г) `new Exception();`
- 8 Создание делегата в C# происходит следующим образом
- А) делегат `MyDelegate();`
  - Б) делегат `MyDelegate();`
  - В) функция `MyDelegate();`
  - Г) событие `MyDelegate();`
- 9 Оператор "as" в C# предназначен для того чтобы
- А) проверка того, является ли объект экземпляром определенного класса или интерфейса
  - Б) приведение объекта к определенному типу данных
  - В) преобразование строки в число
  - Г) проверка равенства двух переменных
- 10 Разница между циклами "while" и "do-while" в C# заключается в том что
- А) `break` выходит из цикла, `continue` пропускает текущую итерацию
  - Б) `break` пропускает текущую итерацию, `continue` выходит из цикла
  - В) и `break`, и `continue` выходят из цикла
  - Г) и `break`, и `continue` пропускают текущую итерацию
- 11 Создание общего класса в C# происходит следующим образом
- А) `class MyClass<T>`
  - Б) `class MyClass<тип T>`
  - В) `class MyClass`

Г) `class MyClass<generic T>`

12 Оператор "lock" в C# предназначен для

А) блокировки потока и предотвращения доступа других потоков к общему ресурсу  
Б) того чтобы разблокировать поток и разрешить другим потокам доступ к общему ресурсу

В) создания нового потока

Г) завершения потока

13 Разница между ключевым словом "var" и явными типами данных в C# заключается в том что

А) var - это ключевое слово, которое указывает компилятору на вывод типа данных, в то время как явные типы данных определяются разработчиком

Б) var используется для переменных, тип которых неизвестен во время компиляции, а явные типы данных используются для переменных, тип которых известен во время компиляции

В) var используется только для примитивных типов данных, тогда как явные типы данных могут использоваться для всех типов данных

Г) разницы нет, var и явные типы данных могут использоваться взаимозаменяемо

14 Определение свойства в C# имеет следующий синтаксис

А) `int MyProperty { get; set; }`

Б) `property int MyProperty { get; set; }`

В) `MyProperty { get; set; }`

Г) `int getMyProperty() { } int setMyProperty(value) { }`

15 Разница между блоком "try-catch" и блоком "try-finally" в C# заключается в том что

А) try-catch используется для обработки ошибок, а try-finally - для задач очистки

Б) -catch обрабатывает исключения и продолжает выполнение, а try-finally только запускает блок finally и продолжает выполнение

В) try-catch выполняет блок catch и выходит из метода, а try-finally только выполняет блок finally и выходит из метода

Г) try-catch и try-finally взаимозаменяемы и выполняют одно и то же действие

16 Создание структуры в C# имеет следующий синтаксис

А) `struct MyStruct { }`

Б) `class MyStruct { }`

В) `object MyStruct { }`

Г) `MyStruct { }`

17 Создание перечисления в C# имеет следующий синтаксис

А) `enum MyEnum { }`

Б) `MyEnum { }`

В) `enum MyEnum { Value1, Value2, Value3 }`

Г) `MyEnum { Value1, Value2, Value3 }`

18 Разница между "абстрактными" и "виртуальными" методами в C# заключается в том, что

А) абстрактные методы не могут иметь реализации, в то время как виртуальные методы могут иметь реализацию

Б) виртуальные методы не могут иметь реализацию, в то время как абстрактные методы могут иметь реализацию

В) абстрактные методы переопределяются по умолчанию, а виртуальные - нет

Г) виртуальные методы переопределяются по умолчанию, а абстрактные – нет

19 Разница между параметрами "out" и "ref" в C# заключается в том, что

А) параметры out должны быть инициализированы перед передачей в метод, параметры ref - нет

Б) параметры ref должны быть инициализированы перед передачей в метод, параметры out - нет

В) параметры out могут использоваться только с типами значений, параметры ref могут использоваться с любыми типами

Г) параметры ref могут использоваться только с типами значений, параметры out могут использоваться с любым типом

20 Разница между переменными "const" и "readonly" в C#

А) переменные const могут быть установлены только во время компиляции, в то время как переменные readonly могут быть установлены во время выполнения программы

Б) переменные readonly могут быть установлены только во время компиляции, в то время как переменные const могут быть установлены во время выполнения программы

В) переменные const можно использовать с любым типом данных, а переменные readonly - только со ссылочными типами

Г) переменные readonly могут использоваться с любым типом данных, а переменные const - только со ссылочными типами

21 Разница между "static" и "instance" методами в C#

А) статические методы могут быть вызваны без создания экземпляра класса, в то время как методы экземпляра требуют наличия экземпляра

Б) методы экземпляра могут быть вызваны без создания экземпляра класса, в то время как статические методы требуют наличия экземпляра

В) статические методы можно использовать только со статическими переменными, тогда как методы экземпляра можно использовать с переменными любого типа

Г) методы экземпляра могут использоваться только с переменными экземпляра, в то время как статические методы могут использоваться с переменными любого типа

22 Термин "межплатформенное программирование" обозначает

А) Программирование для разных операционных систем и платформ

Б) Программирование на нескольких языках программирования

В) Программирование на одной платформе

Г) Программирование только для мобильных устройств

*Выберите несколько правильных ответов*

23 Следующие из перечисленных языков программирования поддерживают межплатформенную разработку с использованием Xamarin

А) JavaScript

Б) F#

В) Java

Г) C#

24 Следующие из нижеперечисленных фреймворков используются для разработки кроссплатформенных мобильных приложений

- A) React Native
- Б) Flutter
- В) .NET Core
- Г) Django

25 Паттерны проектирования, которые широко используются в межплатформенной разработке

- A) MVVM (Model-View-ViewModel)
- Б) Factory Method
- В) Singleton
- Г) Iterator

26 Компоненты, которые часто включаются в межплатформенный стек разработки мобильных приложений

- A) UI фреймворк
- Б) База данных
- В) Только серверная часть
- Г) Веб-сервер

27 Форматы данных, которые часто используются в межплатформенной разработке для обмена информацией между сервером и клиентом

- A) JSON
- Б) XML
- В) MP3
- Г) PNG

28 Плюсы, которые предоставляет межплатформенная разработка по сравнению с нативной разработкой

- A) Экономия времени и ресурсов
- Б) Возможность обслуживать несколько платформ с одним кодом
- В) Только большая производительность
- Г) Отсутствие необходимости в тестировании

29 Компоненты, которые входят в архитектуру многоплатформенного мобильного приложения

- A) Клиентская часть (UI, бизнес-логика)
- Б) Серверная часть (API, база данных)
- В) Только клиентская часть (UI, бизнес-логика)
- Г) Только серверная часть (API, база данных)

30 Следующие из нижеперечисленных инструментов используются для тестирования кроссплатформенных мобильных приложений

- A) Appium
- Б) Xamarin.UITest
- В) JUnit
- Г) PyTest

31 Плюсы, которые могут быть связаны с использованием многоплатформенных инструментов разработки

- A) Реиспользование кода для различных платформ
- Б) Сокращение времени разработки

- В) Ограничение в выборе аппаратных платформ
- Г) Обязательная зависимость от языка программирования

32 Следующие из нижеперечисленных инструментов могут помочь в создании кросс-платформенных десктопных приложений

- А) Electron
- Б) Qt
- В) Xcode
- Г) Android Studio

### Сложные (3 уровень)

33 Инструмент, который позволяет разрабатывать межплатформенные приложения с использованием C#

- А) Xamarin
- Б) Android Studio
- В) Swift
- Г) Eclipse

34 Языки программирования, которые могут быть использованы в межплатформенной разработке

- А) Различные языки программирования
- Б) Только C#
- В) Только Java
- Г) Только Python

35 API в контексте межплатформенного программирования это

- А) Программный интерфейс приложения
- Б) Аппаратный интерфейс устройства
- В) Программа для мультимедийных приложений
- Г) Обновление операционной системы

### Задания на установление соответствия

*Установите соответствие между левым и правым столбцами.*

#### Простые (1 уровень)

- |   |                              |
|---|------------------------------|
| 36 Установите соответствие  |                              |
| 1 Процесс отображения графики на экране или печати  | А) Рендеринг                 |
| 2 Специфический период времени, в течение которого компонент активен и может реагировать на события | Б) Жизненный цикл компонента |
| 3 Процесс адаптации приложения под различные региональные и языковые особенности                    | В) Локализация               |
| 4 Уникальный идентификатор, используемый для аутентификации при взаимодействии с внешними           | Г) API ключ                  |

сервисами

Д) Интеграция

37 Установите соответствие

1 Позволяет создавать объекты без указания конкретных классов создаваемых объектов

А) Фабричный метод

2 Определяет зависимость "один ко многим" между объектами таким образом, что при изменении состояния одного объекта все зависящие от него оповещаются и обновляются автоматически

Б) Наблюдатель

3 Определяет семейство алгоритмов, инкапсулирует каждый из них и делает их взаимозаменяемыми

В) Стратегия

4 Позволяет добавлять новое поведение или состояние объектам во время выполнения

Г) Декоратор

Д) Инкремент

### Средне-сложные (2 уровень)

38 Установите соответствие

1 Xamarin

А) Разработка мобильных приложений на языке C#

2 React Native

Б) Кросс-платформенная библиотека для разработки интерфейсов на JavaScript и React

3 Flutter

В) Фреймворк для создания мобильных приложений с использованием языка Dart

Г) Фреймворк для разработки графических интерфейсов на языке Java

39 Установите соответствие

1 API

А) Интерфейс, предоставляемый программой для взаимодействия с другими программами

2 Сборка мусора

Б) Процесс автоматического освобождения неиспользуемой памяти в программе

3 Полиморфизм

В) Возможность объектов разных классов использовать одинаковые методы или свойства

Г) Возможность выполнения асинхронных операций в программе

40 Установите соответствие

1 async

А) Модификатор метода, указывающий, что он является асинхронным

2 await

Б) Метод, позволяющий асинхронно ожидать завершения выполнения асинхронного метода

3 Task

В) Объект, представляющий асинхронную операцию

Г) Асинхронная операция, возвращающая результат

41 Установите соответствие

- |  |  |
|--|--|
| 1 Целочисленный тип данных, хранящий значения от -2,147,483,648 до 2,147,483,647 | A) int   |
| 2 Тип данных для хранения текстовых строк  | B) string  |
| 3 Логический тип данных, принимающий значения true или false                     | B) bool<br>Г) double   |
| 42 Установите соответствие   |  |
| 1 Thread   | A) Позволяет создать новый поток выполнения в приложении   |
| 2 Task   | B) Предоставляет механизмы для асинхронного выполнения операций и возврата результатов   |
| 3 ThreadPool   | B) Пул потоков, предоставляющий группу потоков для выполнения задач<br>Г) Специальные ключевые слова, упрощающие асинхронное программирование в C#                         |
| 43 Установите соответствие   |  |
| 1 Portable Class Libraries (PCL)   | A) Позволяют определить общий код для использования в разных платформах  |
| 2 Shared Projects  | B) Позволяют создавать платформоспецифичный код для каждой целевой платформы   |
| 3 .NET Standard  | B) Устанавливают стандарты совместимости для библиотек в различных платформах<br>Г) Обеспечивают возможность реализации платформоспецифичного кода в проекте               |
| 44 Установите соответствие   |  |
| 1 Кросс-платформенность  | A) Способность приложения работать на разных операционных системах и устройствах без изменения кода  |
| 2 Реиспользование кода   | B) Использование общего кода для разработки приложений для различных платформ  |
| 3 Адаптивный интерфейс   | B) Создание интерфейса, который адаптируется к разным размерам экранов и устройствам<br>Г) Инструменты, которые позволяют разрабатывать приложения для нескольких платформ |

### Сложные (3 уровень)

- |                               |   |
|-------------------------------|---|
| 45 Установите соответствие    |   |
| 1 MVC (Model-View-Controller) | A) Разделяет приложение на три компонента: модель, представление и контроллер   |
| 2 MVVM (Model-View-ViewModel) | B) Предоставляет отдельный объект (ViewModel) для управления представлением   |
| 3 MVP (Model-View-Presenter)  | B) Разделяет приложение на три компонента: модель, представление и презентер<br>Г) Схема управления состоянием приложения, в которой данные однонаправленно потоки от |

### **Задания открытого типа**

#### **Задания на дополнение**

*Напишите пропущенное слово.*

#### **Простые (1 уровень)**

46 Язык \_\_\_ - это современный объектно-ориентированный язык программирования, используемый для разработки широкого спектра приложений, включая настольные приложения, веб-приложения, мобильные приложения, игры и многое другое

47 Неявно типизированная переменная, ее тип определяется компилятором во время компиляции на основе выражения, используемого для ее инициализации

48 Динамически типизированная переменная, ее тип определяется во время выполнения, что делает ее более гибкой, но и менее безопасной

49 Оператор \_\_\_ в C# используется для указания блока кода, которому требуется определенный ресурс, например, файл или соединение с базой данных. Ресурс автоматически утилизируется после завершения выполнения блока кода

50 Исключения в C# можно перехватывать. Код, который может вызвать исключение, помещается в блок \_\_\_, а код обработки исключения – в блок \_\_\_\_\_. (Перечислить ответы по очереди через пробел)

51 \_\_\_ - это коллекция элементов фиксированного размера одного типа

52 \_\_\_ - это динамически изменяемая коллекция элементов, элементы которой могут быть любого типа

#### **Средне-сложные (2 уровень)**

53 При помощи \_\_\_ в C# можете создать новый класс, который является производным от существующего класса, который содержит в себе все члены базового класса, а также может добавлять новые члены или переопределять существующие

54 В C# схема классов, определяющая набор методов, свойств и событий называется \_\_\_\_\_. Используется для определения общего набора моделей поведения, которые могут быть реализованы несколькими классами, независимо от деталей их реализации

55 Тип в C#, который используется для передачи методов в качестве аргументов другим методам, что позволяет динамически вызывать методы и обрабатывать события называется \_\_\_\_\_.

56 \_\_\_ в C# - это тип значений, который представляет собой именованный постоянный набор значений. Используется для объявления набора именованных постоянных значений, что делает код более читабельным и удобным для сопровождения



57 `__` в C# - это специальный метод, который выполняется при создании объекта из класса. Используется для инициализации состояния объекта при его создании

58 `__` в C# - это член, представляющий именованное значение или атрибут объекта. Используется для инкапсуляции данных объекта и обеспечивает возможность контролируемого доступа и изменения данных

59 `__` в C# может быть объявлено с помощью следующего синтаксиса: «`public event EventHandler EventName;`».

60 Ключевое слово `__` в C# используется для обозначения отсутствия значения или нулевой ссылки. Оно часто используется для инициализации переменных или объектов, которым еще не присвоено значение

61 `__` метод в C# - это метод без имени, который может быть определен и передан в качестве аргумента другому методу. Методы используются для создания встроенных экземпляров делегатов и упрощают код, уменьшая необходимость в отдельных именованных методах

62 `__` в C# - это тип, определяющий ссылку на метод. Используются для передачи методов в качестве аргументов другим методам, что позволяет динамически вызывать методы. Используется для обработки событий и обратных вызовов, а также для других целей

63 Аспект межплатформенного программирования, связанный с общими интерфейсами

64 `__` класс в C# - это класс, который разделен на несколько файлов, но при компиляции рассматривается как один класс. Используется для разделения определения класса на несколько файлов, что облегчает работу над большими классами и упорядочивает код

65 `__` метода в C# - это метод, который имеет то же имя, что и другой метод, но имеет другое количество или типы параметров. Используется для обеспечения нескольких реализаций метода, которые могут быть вызваны на основе аргументов, переданных методу

66 `__` в C# - это структура данных, которая содержит набор элементов. Общие типы `__` в C# включают List, Dictionary, Queue, Stack и Array

67 `__` метод в C# - это метод, который может выполняться асинхронно, что позволяет выполнять длительные задачи без блокировки вызывающего потока. Метод используется для повышения производительности и отзывчивости приложений путем разгрузки трудоемких задач на отдельные потоки

68 `__` в C# - это тип значения, который может содержать члены данных и методы. Похожа на классы, но у нее есть несколько ключевых отличий, например, хранится в стеке, а не в куче, и является типом значения, а не типом ссылок. Это означает, что она хранится непосредственно в памяти, в то время как классы хранятся в куче и доступ к ним осуществляется через ссылки

69 \_\_\_ тип в C# - это тип, которому можно присвоить значение или установить его в null. Этот тип используется для представления значения, которое может отсутствовать

70 \_\_\_ классы обеспечивают переносимость интерфейсов между разными платформами в межплатформенной разработке

71 \_\_\_ структуры данных C# предпочтительны для эффективной обработки больших объемов данных в межплатформенных приложениях

72 Механизм, обеспечивающий безопасность данных в межплатформенных приложениях

73 методы автоматизированного тестирования, которые обеспечивают проверку совместимости межплатформенных приложений с различными устройствами

74 Термин означает "кросс-платформенная разработка"

75 под "MVP (Model-View-Presenter)" в архитектуре программного обеспечения подразумевается

76 Методология "Agile" включает в себя следующие принципы программного обеспечения

### **Сложные (3 уровень)**

77 "паттерн Singleton" в объектно-ориентированном программировании представляет собой

78 Характеристики, которые определяют "реактивное программирование" в современных приложениях

79 "RESTful API" веб-разработке представляют собой

80 Паттерны проектирования, которые связаны с "архитектурой микросервисов"

### **Задания свободного изложения**

*Напишите развернутый ответ в свободной форме, изложив основные положения, факты, применив важнейшие понятия и сделав обобщение по теме задания.*

### **Простые (1 уровень)**

81 Разница между массивом и списком в C# заключается в...

### **Средне-сложные (2 уровень)**

82 Разница между структурой и классом в C# заключается в...

83 Разница между операторами continue и break в C# заключается в...

84 Доступ к статическому члену в C# можно получить, используя...

### **Сложные (3 уровень)**

85 LINQ в C# и как он используется

### Карта учета тестовых заданий

Компетенция	ПК-5: Способность программировать и разрабатывать прикладное программное обеспечение				
Дисциплина	Межплатформенное программирование				
Уровень освоения	Тестовые задания			Задания свободного изложения	Итого
	Закрытого типа		Открытого типа		
	Альтернативный выбор	Установление соответствия/последовательности	На дополнение		
1.1.1 (20%)	5	2	7	1	15
1.1.2 (70%)	27	7	24	3	61
1.1.3 (10%)	3	1	4	1	9
Итого:	35 шт.	10 шт.	35 шт.	5 шт.	85 шт.

### Критерии оценивания

#### Критерии оценивания тестовых заданий

Критерии оценивания: правильное выполнение одного тестового задания оценивается 1 условным баллом, неправильное – 0 баллов.

Максимальная общая сумма баллов за все правильные ответы составляет наивысший балл – 100 баллов.

#### Шкала оценивания результатов компьютерного тестирования обучающихся (рекомендуемая)

Оценка	Процент верных ответов	Баллы
«удовлетворительно»	70-79%	61-75 баллов
«хорошо»	80-90%	76-90 баллов
«отлично»	91-100%	91-100 баллов

№ тестовых заданий	Номер и вариант правильного ответа
1	Б
2	А
3	А
4	А
5	А
6	А
7	А
8	А
9	А
10	А
11	А
12	А
13	А
14	А
15	А
16	А
17	А
18	А
19	А
20	А
21	А
22	А
23	А, Б
24	А, Б
25	А, Б
26	А, Б
27	А, Б
28	А, Б
29	А, Б
30	А, Б
31	А, Б
32	А, Б
33	А
34	А
35	А
36	1-А, 2-Б, 3-В, 4-Г

## Ключи ответов

46	С#, с#
47	var
48	dynamic
49	using
50	try catch, catch try
51	Массив
52	Список
53	наследования, наследование
54	интерфейсом, интерфейс
55	делегатом, делегат, delegate
56	Перечисление
57	Конструктор
58	Свойство
59	Событие
60	null
61	Анонимный
62	Делегат
63	Полиморфизм
64	Частичный
65	Перегрузка
66	Коллекция
67	Асинхронный
68	Структура
69	Нулевой
70	Абстрактные, Абстрактный
71	Коллекции
72	Шифрование
73	Совместимость
74	Универсальность
75	Представление
76	Итерации
77	Единственный экземпляр
78	Потоки данных
79	Маршруты
80	Декомпозиция
81	Массив - это коллекция элементов фиксированного размера одного типа, а список - это динамически изменяемая коллекция элементов, которые могут быть любого типа
82	Структура – это тип значений, в то время как класс – это ссылочный тип. Это означает, что когда struct передается как аргумент метода или присваивается переменной, создается копия struct, в то время как передается или присваивается ссылка на исходный объект класса

<b>37</b>	1-А, 2-Б, 3-В, 4-Г
<b>38</b>	1-А, 2-Б, 3-В
<b>39</b>	1-А, 2-Б, 3-В
<b>40</b>	1-А, 2-Б, 3-В
<b>41</b>	1-А, 2-Б, 3-В
<b>42</b>	1-А, 2-Б, 3-В
<b>43</b>	1-А, 2-Б, 3-В
<b>44</b>	1-А, 2-Б, 3-В
<b>45</b>	1-А, 2-Б, 3-В

<b>83</b>	Оператор continue пропускает текущую итерацию цикла и переходит к следующей итерации, а оператор break завершает цикл и выходит из структуры цикла
<b>84</b>	Доступ к статическому члену в С# можно получить, используя имя класса и оператор точки (.), за которым следует имя члена. Например, для доступа к статическому свойству MyProperty в классе MyClass используется следующий синтаксис: MyClass.MyProperty
<b>85</b>	LINQ (Language Integrated Query) в С#\# - это набор языковых и библиотечных функций, которые позволяют выполнять запросы к данным из различных источников данных. LINQ используется для выполнения таких операций, как фильтрация, сортировка и преобразование данных в краткой и выразительной форме

