

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Тверской государственный технический университет»  
(ТвГТУ)

УТВЕРЖДАЮ

Проректор  
по научной работе



Е.А. Евстифеева  
2016 г.

**ПРОГРАММА  
ВСТУПИТЕЛЬНОГО ИСПЫТАНИЯ  
(МЕЖДИСЦИПЛИНАРНОГО ЭКЗАМЕНА)  
И ПОРЯДОК ЕГО ПРОВЕДЕНИЯ**

**для поступающих в магистратуру**

**по направлению подготовки**

**09.04.04 Программная инженерия**

**Направленность (профиль) подготовки**

**Разработка программно-информационных систем**

Вступительное испытание проводится в форме **письменного экзамена**

Тверь 2016

Программа содержит перечень тем (вопросов) по дисциплинам базовой и (или) вариативной части учебного плана подготовки бакалавров по \_\_\_\_\_  
09.03.04 Программная инженерия

(Код и наименование направления, направленность (профиль))

вошедших в содержание билетов (тестовых заданий) вступительных испытаний в магистратуру.

Составители:

д.ф.-м.н., профессор

к.т.н., доцент

к.т.н., доцент

Калабин А.Л.

Биллиг В.А.

Мальков А.А.

Программа обсуждена и рекомендована к использованию на заседании кафедры Программное обеспечение (протокол № 11 от 6 июня 2016 г.).

Научный руководитель образовательной программы  
магистерской подготовки к.т.н., доцент

Мальков А.А.

Зав. кафедрой

д.ф.-м.н., профессор

Калабин А.Л.

Согласовано:

Начальник УАР

С.В. Рассадин

## **1. ДИСЦИПЛИНЫ, ВКЛЮЧЁННЫЕ В ПРОГРАММУ ВСТУПИТЕЛЬНЫХ ИСПЫТАНИЙ В МАГИСТРАТУРУ**

1.1. Технология разработки ПО АС в промышленности.  
Унифицированный язык моделирования UML

1.2. Теория языков программирования и методов трансляции

1.3. Технология разработки программного обеспечения. Объектно-ориентированное программирование

## **2. СОДЕРЖАНИЕ УЧЕБНЫХ ДИСЦИПЛИН**

2.1. Технология разработки ПО АС в промышленности.  
Унифицированный язык моделирования UML

Темы (вопросы)

1. Назначение языка UML. Общая структура языка UML. Характеристики основных объектов.

2. Базовые принципы языка UML. Абстракция. Наследование. Полиморфизм. Инкапсуляция. Передача сообщений. Ассоциации. Агрегирование. Зависимость.

3. Семантика и синтаксис UML. Нотация UML. Типы диаграмм UML.

4. Особенности изображения диаграмм языка UML. Диаграмма Вариантов использования (UseCaseDiagram). Диаграмма Классов (ClassDiagram). Диаграмма Состояний (Statechartdiagram). Диаграмма Последовательности (sequencediagram).

5. Особенности изображения диаграмм языка UML. Диаграмма Кооперации (Collaborationdiagram). Диаграмма Компонентов (componentdiagram). Диаграмма развёртывания (deploymentdiagram).

6. Этапы создания программных продуктов. Структура программных продуктов. Жизненный цикл ПО. Модели жизненного цикла ПО.

7. Основы методологии разработки (проектирования) программного обеспечения. Использование стандартов и методологий в жизненном цикле разработки и сопровождения программного обеспечения информационных систем.

8. Структурный подход к проектированию программного обеспечения. Методология функционального моделирования SADT. Моделирование данных. Методология IDEF1X.

9. RationalUnifiedProcess - полный технологический процесс, адаптируемый к условиям организации-заказчика и его гармонизация с ГОСТ Р ИСО/МЭК 1220714. Методология RAD.

10. Документирование ПО. Обзор отечественных стандартов по составлению документации на ПО.

11. Информационные технологии и архитектура промышленных предприятий. Инструменты разработки архитектуры. Основные модели и инструменты описания бизнес-архитектуры.

12. Основные модели описания архитектуры автоматизированных систем. Матрица Захмана.

13. Особенности проектов информационных систем. История возникновения CASE-средств.

14. Инструментальные средства для разработки и сопровождения архитектуры предприятия. CASE-средства разработки проектов информационных систем. Общая характеристика и классификация.

15. Анализ бизнеса с различных сторон: поддержка в BPrwin нотаций: IDEF0, IDEF3 и DFD. Характеристика инструментального средства AllFusionERwinDataModeler (ERwin).

16. Оценка и выбор CASE-средств. Критерии оценки и выбора. Технология внедрения CASE-средств. Определение критериев успешного внедрения. Выполнение пилотного проекта. Оценка пилотного проекта.

#### Литература для подготовки

1. Орлов, С.А. Технологии разработки программного обеспечения: разработка сложных программных систем; учеб. пособие для вузов по напр. подготовки бакалавров и магистров по спец. "Информатика и вычисл. техника" - М. [и др.]: Питер, 2002. - 464 с. - (11278-3) (681; О-66)

2. Якобсон, А. Унифицированный процесс разработки программного обеспечения / Якобсон, А., Буч, Г., Рамбол, Дж. - СПб. [и др.]: Питер, 2002. - 492 с. - (10019-1) (681; Я 46)

3. Кинг Д. Создание эффективного программного обеспечения / под ред. В.В. Мартынюка - М.: Мир, 1991. - 287 с. - (49058-1) (681; К 41)

4. Применение современных языков и инструментов для моделирования предметной области автоматизации [Текст]: учеб. пособие. Ч. 1 / Палюх, Б.В., Котлинский, С.В., Ключин, А.Ю. ; Тверской гос. техн. ун-т - Тверь: ТвГТУ, 2013. - 187 с. - (99446-72) (004; П14)

5. Применение современных языков и инструментов для моделирования предметной области автоматизации [Текст]: учеб. пособие. Ч. 2 / Палюх, Б.В., Котлинский, С.В., Ключин, А.Ю. ; Тверской гос. техн. ун-т - Тверь: ТвГТУ, 2013. - 172 с. - (99447-72) (004; П14)

6. Гецци, К. Основы инженерии программного обеспечения [Текст] / Гецци, К., Джазайери, М., Мандриоли, Д. - СПб.: БВХ-Петербург, 2005. - 805 с. - (59148-1) (681; Г 45)

7. Липаев, В.В. Программная инженерия. Методологические основы [Текст]: учеб. для вузов по напр. "Бизнес-информатика" (080700) / Гос. ун-т высш. шк. экономики - М.: ТЕИС, 2006. - 607 с. - (75271-1) (681; Л 61)

2.2. Теория языков программирования и методов трансляции  
Темы (вопросы)

1. Структура компилятора. Типы транслируемых программ.
2. Определение языка. Синтаксис и семантика.
3. Порождающие грамматики Хомского. Примеры порождающих грамматик.
4. Классификация грамматик. Иерархия Хомского.
5. Контекстно-свободные грамматики.
6. Деревья вывода. Канонические выводы. Двусмысленные порождающие грамматики
7. Проблема разбора.
8. Лексический анализ.
9. БНФ-нотация и контекстно-свободные грамматики.
10. LL(n) и LR(n) – грамматики.
11. Построение синтаксического графа.
12. Построение программы грамматического разбора для заданного синтаксиса.
13. Восходящий синтаксический анализ.
14. Работа с таблицей символов.
15. Восстановление при синтаксических ошибках.
16. Постфиксная запись.
17. Методы генерирования кода.
18. Грамматики арифметических выражений. Перевод арифметических выражений в ПОЛИЗ.

Литература для подготовки

1. И.Г. Кревский, М.Н. Селиверстов, К.В. Григорьева **ФОРМАЛЬНЫЕ ЯЗЫКИ, ГРАММАТИКИ И ОСНОВЫ ПОСТРОЕНИЯ ТРАНСЛЯТОРОВ**

Учебное пособие (под ред. д.т.н., профессора А.М. Бершадского) Пенза  
2003

2. Свердлов С. З. Языки программирования и методы трансляции: Учебное пособие. — СПб.: Питер, 2007. — 638 с: ил.
3. Карпов Ю.Г. Методы построения трансляторов СПб.: Питер, 2001. – 113 с.

### 2.3. Технология разработки программного обеспечения. Объектно-ориентированное программирование

Темы (вопросы)

1. Основные принципы и этапы объектно-ориентированного проектирования и программирования.
2. Классы и объекты.
3. Представление объектов в памяти. Стек и куча. Значимые и развернутые типы.
4. Структуры.
5. Две роли класса. Класс как тип данных. Класс как модуль.
6. Отношения между классами. Наследование. Множественное наследование. Интерфейсы
7. Отношения между классами. Отношение вложенности - "клиент - поставщик".
8. Классы, задающие функциональный тип. Делегаты.
9. Классы с событиями.
10. Перечисления. Шкалы.
11. Классы с родовыми параметрами. Универсализм. Ограничение универсальности.
12. Атрибуты и теги. Документирование проекта.
13. Исключительные ситуации. Отладка.
14. Тестирование проекта.
15. Корректность класса.
16. Стиль программирования.

Литература для подготовки (доступна на сайте Интернет университета)

1. Бертран Мейер "Объектно-ориентированное конструирование программных систем", М. 2005 г. Изд. "Русская Редакция", "Интернет Университет"
2. Бертран Мейер "Почувствуй класс. Учимся программировать хорошо с объектами и контрактами", М. 2011 г. Изд. "Бином. Лаборатория знаний", Национальный открытый университет ИНТУИТ"

3. Биллиг В. А. "Основы объектного программирования на С#" М. 2010 г. Изд. "Бином. Лаборатория знаний", Национальный открытый университет ИНТУИТ"

### 3. ПРИМЕР ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тверской государственный технический университет»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № \_\_\_\_\_

вступительных испытаний для абитуриентов направления подготовки  
магистратуры 09.04.04 Программная инженерия  
Направленность (профиль) «Разработка программно-  
информационных систем»

#### БЛОК 1

##### 1. Вопрос 1

Необходимо поместить метаданные, созданные с помощью `sn.exe` –к `c:\myStrongName.key`, в сборку для указания "строного имени" приложения. Какой фрагмент кода следует использовать в `cs`-файле?

1. `[class: AssemblyKeyFile("c:\myStrongName.key")]`
2. `[property: AssemblyKeyFile("c:\myStrongName.key")]`
3. `[module: AssemblyKeyFile("c:\myStrongName.key")]`
4. `[all: AssemblyKeyFile("c:\myStrongName.key")]`
5. `[assembly: AssemblyKeyFile("c:\myStrongName.key")]`

##### Вопрос 2

В строке `string parameters` содержатся названия параметров (`Server`, `Host`, `Register`), а в строке `string values` – их значения (`www.my.com`, `my.net`, `"No"`). Вам необходимо записать значение параметра `Server` в строку `string server`. Каким фрагментом кода следует воспользоваться для решения поставленной задачи?

1. `string server = parameters+values; server=server.Substring("Server",0)`
2. `string[] tmp = values.Split(','); string server = tmp[0]`
3. `string server = values[0]`
4. `string server = values.IndexOf("Server",0)`
5. `string tmp = values.Substring(0,','); string server = tmp.Replace(","," ")`

##### Вопрос 3

Вы разработали веб-службу `MyWebService`. Далее Вы хотите создать клиентское приложение, взаимодействующее с созданной веб-службой, для чего необходимо создать класс-посредник.

Какое средство следует использовать для решения поставленной задачи?

1. Инструментальное средство `signcode.exe`
2. Инструментальное средство `ResGen.exe`
3. Инструментальное средство `AxImp.exe`
4. Инструментальное средство `xsd.exe`
5. Инструментальное средство `wSDL.exe`



#### Вопрос 4

Вы разрабатываете класс работы с комплексными числами – класс Complex. Для удобства отображения комплексных чисел Вы хотите добавить в свой класс метод явного преобразования из типа Complex в тип string. Какой заголовок следует применить для этого метода?

1. public explicit operator Complex(string theString)
2. public operator Complex(string theString)
3. public static implicit operator Complex(string theString)
4. public static implicit operator string(Complex theComplex)
5. public static implicit operator int(Complex theComplex)

#### Вопрос 5

Какой из перечисленных протоколов позволяет обращаться к веб-службам, созданным при помощи ASP.NET?

1. UDP
2. FTP
3. SOAP
4. SMTP
5. TCP

#### Вопрос 6

Вы желаете создать строгое имя (Strong Name) для использования в своей сборке. Какие действия следует предпринять для решения поставленной задачи?

1. Создать с помощью утилиты sn.exe -T файл с ключами, затем поместить путь к созданному файлу в атрибут [assembly: AssemblyKeyName(")], расположенный в AssemblyInfo.cs
2. Создать с помощью утилиты ResGen.exe -Z файл с ресурсами, затем поместить путь к созданному файлу в атрибут [assembly: AssemblyCulture(")], расположенный в AssemblyInfo.cs
3. Создать с помощью утилиты wsdl.exe файл с классом-посредником, затем поместить путь к созданному файлу в атрибут [assembly: AssemblyProduct(")], расположенный в AssemblyInfo.cs
4. Создать файл с помощью утилиты sn.exe -k с ключами, затем поместить путь к созданному файлу в атрибут [assembly: AssemblyKeyFile(")], расположенный в AssemblyInfo.cs
5. Создать с помощью утилиты ResGen.exe -i файл с ресурсами, затем поместить путь к созданному файлу в атрибут [assembly: AssemblyConfiguration(")], расположенный в AssemblyInfo.cs

#### Вопрос 7

Вы разрабатываете приложение, рассылающее короткие сообщения на мобильные телефоны. Ваша задача – разместить текстовое сообщение, которое передал Вам оператор в переменной msg типа string, на вершине объекта messages типа Stack. Какую строку следует использовать для решения поставленной задачи?

1. messages.Add(msg)
2. messages.Push(msg)
3. messages.Contains(msg)
4. messages.Pop(msg)
5. messages.Peek(msg)

### Вопрос 8

Василий хочет переопределить полиморфный метод Му. Какое ключевое слово следует использовать Василию при объявлении метода Му?

1. delegate
2. protected
3. sealed
4. override
5. abstract

### Вопрос 9

Алексею поручили написать приложение, считывающее классы, интерфейсы, массивы, значения и перечисления, содержащиеся в библиотеке ядра платформы .NET Framework (mscorlib.dll).

```
using System;
namespace Boo
{
    class C
    {
        public static void Main()
        {
            Assembly my = Assembly.Load("mscorlib.dll");
            Type[] types = my.GetObjectData();
            foreach (Type _type in types)
            {
                Console.WriteLine("Тип: {0}", _type);
            }
            Console.WriteLine("Всего обнаружено {0} типов", types.Length);
        }
    }
}
```

Каким образом возможно устранить ошибку, возникающую при компиляции данного фрагмента кода?

1. Следует полностью указать путь к библиотеке mscorlib.dll. Для использования объекта Types следует подключить System.Data
2. При выводе типа на консоль вместо \_type следует использовать \_type.ToString(). Следует полностью указать путь к библиотеке mscorlib.dll
3. Следует подключить System.Reflection. Неверно использован метод GetObjectData – вместо него следует использовать метод GetTypes()
4. Вместо массива объектов Type[] следует использовать массив объектов Method[]. Соответственно, в цикле foreach нужно заменить Type \_types на Method \_types
5. Вместо объекта типа Assembly следует использовать объект типа class. Вместо неправильно примененного метода GetObjectData нужно использовать метод LoadModule()

### Вопрос 10

Вы разрабатываете почтовый клиент. Вам необходимо, чтобы каждые 5 минут он проверял наличие новой почты. Для реализации этой задачи Вы встроили в главную форму объект myTimer, производный от System.Windows.Forms.Timer.

Какое событие, генерируемое этим объектом, Вы будете использовать?

1. Minute
2. Tick
3. OnTick
4. WM\_TIMER
5. OnWMTimer

## БЛОК 2

Вопрос 1. Назначение языка UML. Общая структура языка UML. Характеристики основных объектов.

Вопрос 2. Объектное программирование. Основные принципы и этапы объектно-ориентированного программирования. Архитектура программы при ООП. Наследование, полиморфизм, композиция, наполнение.

Вопрос 3. Структура и модель описания ИТ-архитектуры Gartner. Методика METAGroup. Модель процесса разработки и использования архитектуры.

Вопрос 4. Классификация грамматик. Иерархия Хомского.

## БЛОК 3

Задача Ады Лавлейса. Вычисление чисел Бернулли

Первая программа для вычислительной машины (машины Бэббиджа) была создана графиней Адой Лавлейс, дочерью поэта лорда Байрона. Ада Лавлейс по праву считается первым в мире программистом. В её честь назван язык программирования «Ада». День её рождения отмечается как день программиста. Программа, которую написала Ада Лавлейс, позволяла считать числа Бернулли, широко применяемые в разных областях математики.

Якоб Бернулли, представитель славной семьи математиков и физиков, ввёл понятие чисел Бернулли, изучая суммы степеней целых чисел. Вот формула, выведенная Якобом Бернулли:

$$\sum_{n=0}^{N-1} n^k = \frac{1}{k+1} \sum_{m=0}^k C_{k+1}^m * B_m * N^{k+1-m}$$

Здесь  $C_{k+1}^m$ , - биномиальный коэффициент, а  $B_m$  – число Бернулли.

Биномиальный коэффициент задаёт число сочетаний  $k$  предметов, выбранных из  $n$  предметов, определяется как:

$$C_n^k = \frac{n!}{k!(n-k)!}$$

*Рекуррентное соотношение*

Числа Бернулли можно рассчитать, используя следующую рекуррентную формулу:

$$B_m = \frac{-1}{m+1} \sum_{k=1}^m C_{m+1}^{k+1} * B_{m-k} \quad (1)$$

Свойства чисел Бернулли:

- $B_0 = 1$ ;
- Все нечётные числа, за исключением  $B_1$  равны 0.
- Чётные числа меняют знак.
- С ростом  $m$  числа Бернулли по модулю возрастают.

Вот как выглядит последовательность первых чисел Бернулли:

$B_0 = 1$ ;  $B_1 = -1/2$ ;  $B_2 = +1/6$ ;  $B_3 = 0$ ;  $B_4 = -1/30$ ;  $B_5 = 0$ ;  $B_6 = +1/42$ ;  $B_8 = -1/30$ ;  $B_{10} = +5/66$ ;  
 $B_{12} = -691/2730$ ;  $B_{14} = +7/6$ ;  $B_{16} = -3617/510$ ;

*Сходящийся ряд*

Значения модулей чётных чисел Бернулли можно вычислять, используя соотношение, включающее быстро сходящийся ряд:

$$B_m = \frac{(2*m)!}{\pi^{2*m} * 2^{2*m-1}} \sum_{i=1}^{\infty} \frac{1}{i^{2*m}} \quad (2)$$

При таком определении последовательность чисел Бернулли выглядит следующим образом:  
 $B_1 = +1/6$ ;  $B_2 = 1/30$ ;  $B_3 = 1/42$ ;  $B_4 = 1/30$ ;  $B_5 = 5/66$ ;  $B_6 = 691/2730$ ;  $B_7 = 7/6$ ;  $B_8 = 3617/510$ ;

### *Задание на программирование*

#### Вопрос 1

Постройте проект, включающий два метода вычисления чисел Бернулли, используя соотношения (1) и (2). Постарайтесь построить оптимальную по времени реализацию методов.

#### Вопрос 2

Проведите сравнительный анализ эффективности методов по времени выполнения.

Научный руководитель образовательной программы  
магистерской подготовки к.т.н., доцент

Мальков А.А.

**БЛАНК ОТВЕТОВ АБИТУРИЕНТА № \_\_\_\_\_**

Бланк ответов блока 1:

№ вопроса	1	2	3	4	5	6	7	8	9	10
Ответ										
Балл (Выставляется членом экзаменационной комиссии)										

Бланк ответов блока 2:

№ вопроса	11	12	13	14
Ответ				
Балл (Выставляется членом экзаменационной комиссии)				
Примечание				

Бланк ответов блока 3:

№ вопроса	15	16
Ответ		
Балл (Выставляется членом экзаменационной комиссии)		
Примечание		