МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМЕНИ И.С. ТУРГЕНЕВА»

Кафедра программной инженерии

**ОТЧЕТ**

По лабораторной работе № 1

на тему: «Знакомство с платформой .NET. Реализация консольных приложений»

по дисциплине: «Проектирование и архитектура программных систем»

Выполнила: Баркова О. А., Евсикова М. Ю.

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 92ПГ

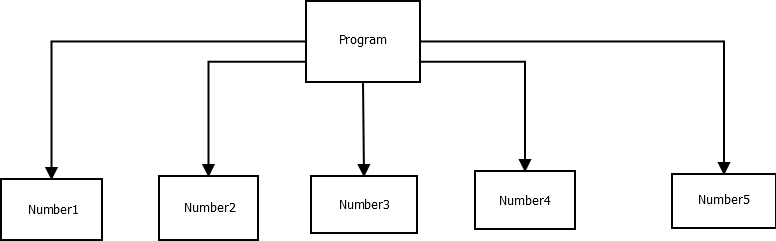
Проверил: Валухов В.А.

Отметка о зачете:

Дата: «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2021 г.

Орел, 2021

**Описание модульной архитектуры**



Данная программа состоит из модулей (подпрограмм), а именно главного и дополнительных. Главный модуль «Program» является связующим для дополнительных. Всего в программе содержится пять дополнительных модулей. Рассмотрим каждый из них отдельно: модуль «Number1» выполняет функцию вывода аргументов из командной строки, «Number2» помогает нам определить какие года в промежутке от 1900 до 2000 являются високосными. Модуль «Number3» выводит числа Фибоначи меньшие введенного нами числа. Четвертый модуль «Number4» выводит факториал заданного числа. Последний модуль «Number5» выполняет алгоритм Решето Эратосфена для нахождения простых чисел от 2 до n, затем выводит полученные значения на экран.

**Ответы на контрольные вопросы.**

**1. Что понимается под термином архитектура программной системы?**

Архитектура - это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Система - это набор компонентов, объединенных для выполнения определенной функции или набора функций.

Архитектура программного обеспечения системы или набора систем состоит из всех важных проектных решений по поводу структур программы и взаимодействий между этими структурами, которые составляют системы. Таким образом архитектура программы или компьютерной системы - это структура или структуры системы, которые включают элементы программы, видимые извне свойства этих элементов и связи между ними.

**2. Перечислите основные разновидности архитектуры программных систем.**

Клиент-серверные системы

Web – сервисы и Web-приложения

Интегрированные распределенные решения (solutions)

Встроенные системы

Системы реального времени

Программное обеспечение мобильных устройств

Программное обеспечение носимых (wearable) компьютеров

Программное обеспечение промежуточного уровня (middleware)

ПО для облачных вычислений и центров обработки данных

ПО для компьютерных кластеров

ПО для виртуализации

ПО для управления информацией (information management)

ПО для управления знаниями (knowledge management)

ПО для научных вычислений (scientific computing)

**3. Какие требования предъявляются к проектированию модульной архитектуры программной системы?**

Модуль — это не произвольный кусок кода, а отдельная функционально осмысленная и законченная программная единица (подпрограмма), которая обеспечивает решение некоторой задачи и в идеале может работать самостоятельно или в другом окружении и быть переиспользуемой.

*1. Иерархическая.*

Сначала систему разбивают на крупные функциональные модули/подсистемы, описывающие ее работу в самом общем виде. Затем, полученные модули, анализируются более детально и, в свою очередь, делятся на под-модули либо на объекты.

*2. Функциональная.*

Деление на модули/подсистемы лучше всего производить исходя из тех задач, которые решает система. Основная задача разбивается на составляющие ее подзадачи, которые могут решаться/выполняться независимо друг от друга. Каждый модуль должен отвечать за решение какой-то подзадачи и выполнять соответствующую ей функцию. Помимо функционального назначения модуль характеризуется также набором данных, необходимых ему для выполнения его функции, то есть:

Модуль = Функция + Данные, необходимые для ее выполнения.

*3. High Cohesion + Low Coupling*

Самым же главным критерием качества декомпозиции является то, насколько модули сфокусированы на решение своих задач и независимы. Обычно это формулируют следующим образом: "Модули, полученные в результате декомпозиции, должны быть максимально сопряженны внутри, т.е. сфокусированы на решении одной узкой проблемы, а не занимаются выполнением (high internal cohesion) и минимально связанны друг с другом, т.е. они должны иметь возможность взаимодействовать, но при этом как можно меньше знать друг о друге (low external coupling)."

Модули должны обладать следующими свойствами:

функциональная целостность и завершенность — каждый модуль реализует одну функцию; модуль самостоятельно выполняет полный набор операций для реализации своей функции.

один вход и один выход — на входе программный модуль получает определенный набор исходных данных, выполняет содержательную обработку и возвращает один набор результатных данных;

логическая независимость — результат работы программного модуля зависит только от исходных данных, но не зависит от работы других модулей;

слабые информационные связи с другими модулями — обмен информацией между модулями должен быть по возможности минимизирован.

**4. Перечислите основные особенности языка программирования C# и платформы .NET.**

*Основные особенности языка C#*

На сегодняшний момент язык программирования C# один из самых мощных, быстро развивающихся и востребованных языков в ИТ-отрасли.

C# является языком с Си-подобным синтаксисом и близок в этом отношении к C++ и Java.

C# является объектно-ориентированным и в этом плане много перенял у Java и С++. Например, C# поддерживает полиморфизм, наследование, перегрузку операторов, статическую типизацию. Объектно-ориентированный подход позволяет решить задачи по построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений. И C# продолжает активно развиваться, и с каждой новой версией появляется все больше интересных функциональностей, как, например, лямбды, динамическое связывание, асинхронные методы и т.д.

*Основные особенности платформы .NET*

*Поддержка нескольких языков.* Основой платформы является общеязыковая среда исполнения Common Language Runtime (CLR), благодаря чему .NET поддерживает несколько языков. Поэтому при определенных условиях мы можем сделать отдельные модули одного приложения на отдельных языках.

*Кроссплатформенность.* .NET является переносимой платформой (с некоторыми ограничениями). Используя различные технологии на платформе .NET, можно разрабатывать приложения на языке C# для самых разных платформ - Windows, MacOS, Linux, Android, iOS, Tizen.

*Мощная библиотека классов.* .NET представляет единую для всех поддерживаемых языков библиотеку классов. И какое бы приложение мы не собирались писать на C# - текстовый редактор, чат или сложный веб-сайт - так или иначе мы задействуем библиотеку классов .NET.

*Разнообразие технологий.* Общеязыковая среда исполнения CLR и базовая библиотека классов являются основой для целого стека технологий, которые разработчики могут задействовать при построении тех или иных приложений. Для построения графических приложений с богатым насыщенным интерфейсом - технология WPF и UWP, для создания более простых графических приложений - Windows Forms. Для разработки мобильных приложений - Xamarin. Для создания веб-сайтов и веб-приложений - ASP.NET и т.д.

*Производительность.* Приложения на .NET 5 отличаются высокой производительностью.

**5. Как происходит компиляция и выполнение программы на языке C#?**

При компиляции код на любом из этих языков компилируется в сборку на общем языке CIL (Common Intermediate Language) - своего рода ассемблер платформы .NET. Код на C# компилируется в приложения или сборки с расширениями exe или dll на языке CIL. Далее при запуске на выполнение подобного приложения происходит JIT-компиляция (Just-In-Time) в машинный код, который затем выполняется. При этом, поскольку наше приложение может быть большим и содержать кучу инструкций, в текущий момент времени будет компилироваться лишь та часть приложения, к которой непосредственно идет обращение. Если мы обратимся к другой части кода, то она будет скомпилирована из CIL в машинный код. При том уже скомпилированная часть приложения сохраняется до завершения работы программы. В итоге это повышает производительность.