

## **Общие методические рекомендации по организации выполнения лабораторных работ и контрольных заданий по учебной дисциплине «Численные Методы»**

Обязательным этапом всех без исключения заданий по данной учебной дисциплине является разработка и реализация вычислительной программы, результатом выполнения которой является определенный набор числовых данных, представленный в заданном формате.

Некоторые из заданий в ходе своего выполнения будут требовать использования ранее уже разработанных алгоритмов, методов, функций или наборов числовых данных.

По этой причине мы воспользуемся возможностью объединения всех разрабатываемых алгоритмов в составе единой динамической библиотеки численных методов.

В свою очередь, алгоритмы, относящиеся к определенным разделам численных методов, например, к разделу «Интерполяция функций», мы будем объединять в рамках отдельных классов, входящих в состав указанной динамической библиотеки.

Такой подход в организации процесса разработки сложных приложений имеет определенные преимущества.

Во-первых, студент учится разрабатывать такие программные компоненты, которые затем могут быть использованы в иных самостоятельных приложениях.

Во-вторых, студент приобретает навыки компоновки исполняемых приложений, когда их составные части уже откомпилированы и размещаются в самостоятельных библиотеках.

В-третьих, студент может разрабатывать приложения, составные части которого реализованы на различных языках программирования, например, на **Fortran**, **C++** и **C#**.

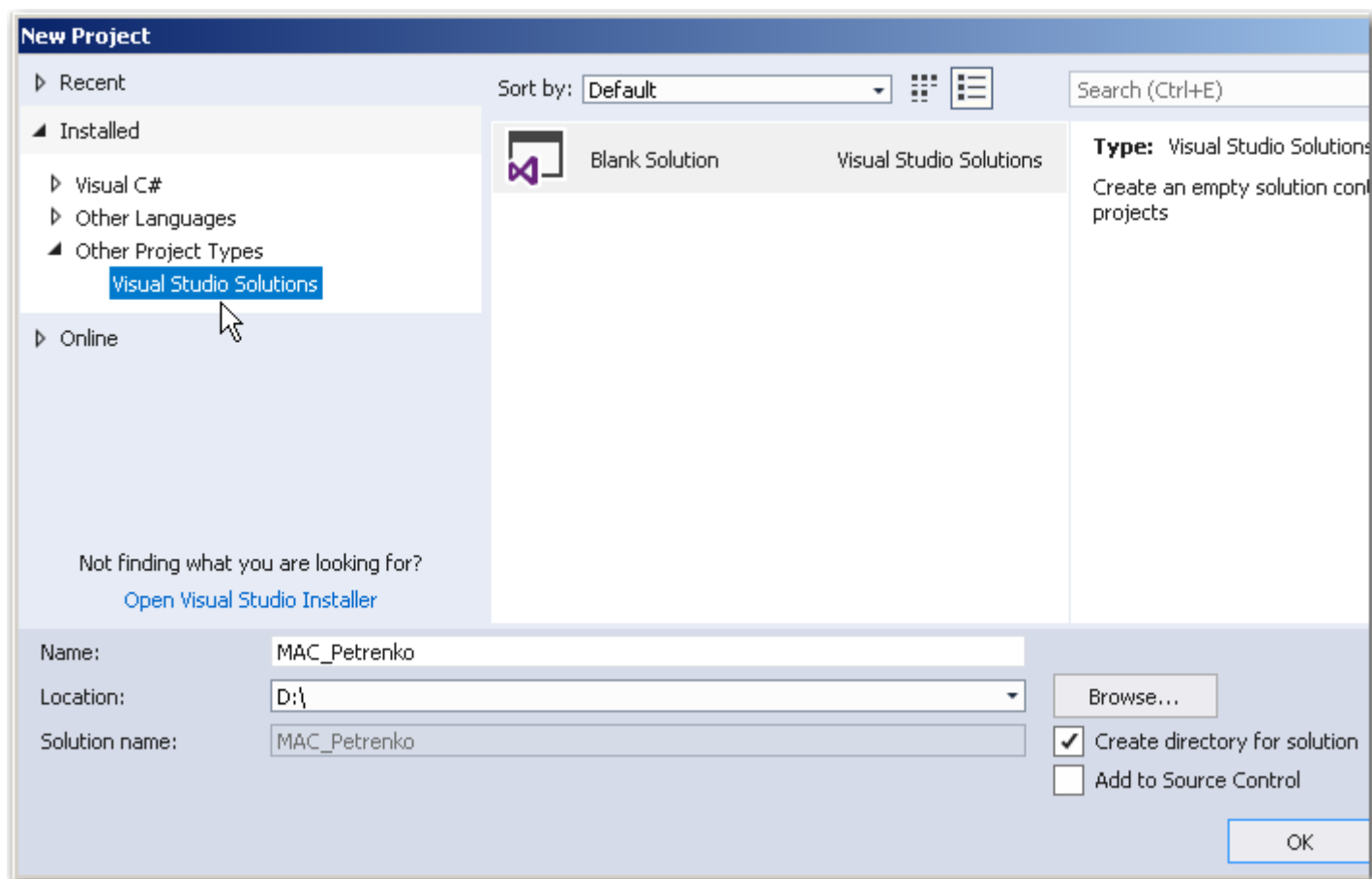
Кроме того, библиотека численных методов, разработанная студентом в рамках данной учебной дисциплины, может быть использована при выполнении курсовой работы по смежной учебной дисциплине или при реализации вычислений в рамках дипломной работы.

Полностью оформленное руководство по динамической библиотеке численных методов является основой **курсовой работы** по дисциплине «**Численные Методы**» и оценивается самостоятельной итоговой оценкой в конце учебного семестра.

Среда разработки приложений, которой мы будем пользоваться при выполнении лабораторных и контрольных заданий, это – **MS Visual Studio (MSVS)**, в состав которой входит полный инструментальный язык программирования **C#**.

Первым этапом работы будет создание рабочего пространства **MSVS – Visual Studio Solution**, предназначенного для упорядоченного (систематизированного) размещения разрабатываемых программ, библиотек и других файлов данных.

Для этого стартуем среду разработки **MSVS** и генерируем в ней новый проект –



Следует обратить внимание на задаваемые параметры:

- тип проекта (**Project types: Other Project Types – Visual Studio Solutions**),
- имя проекта (**Name: MAC\_Petrenko**).

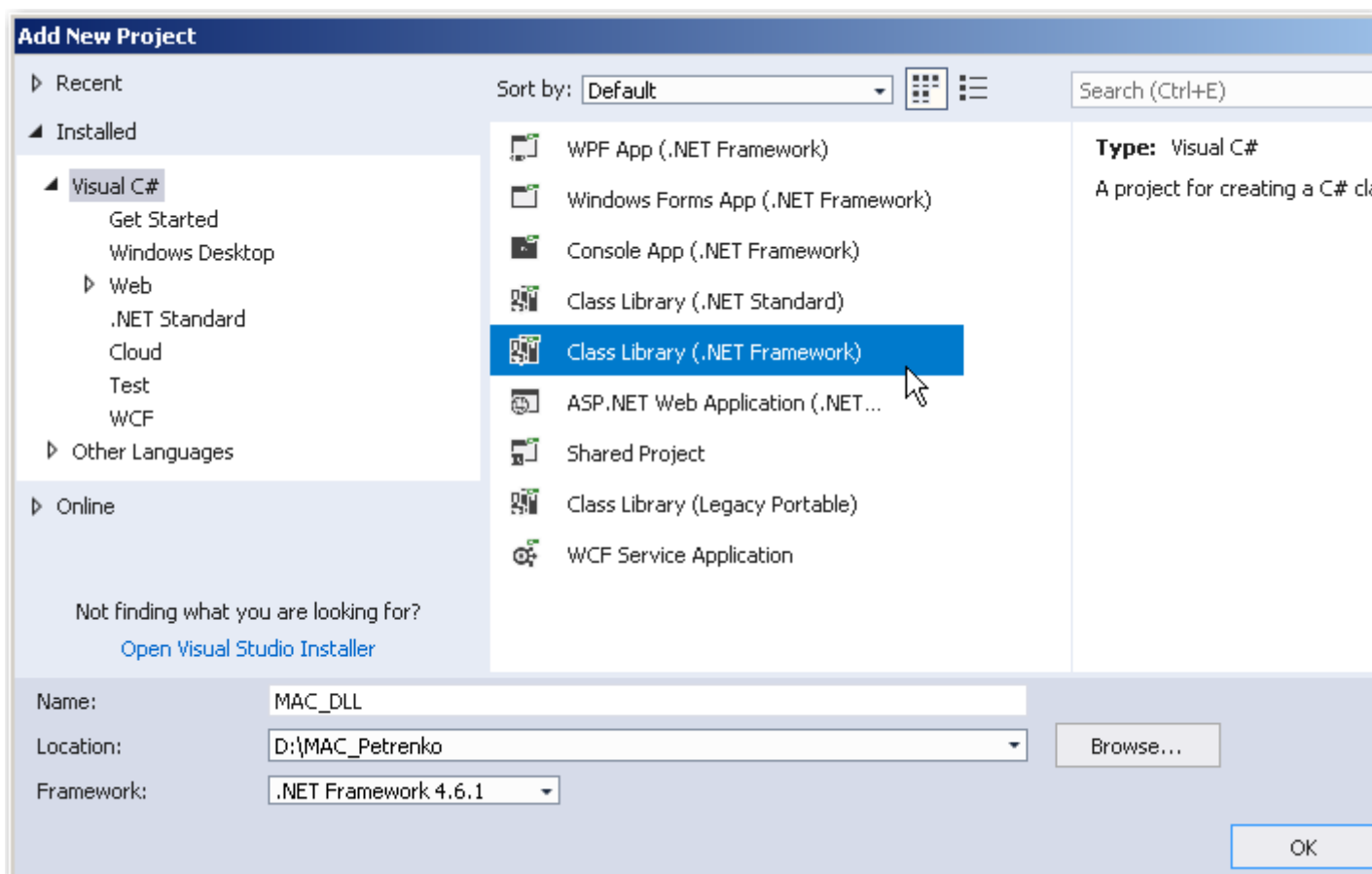
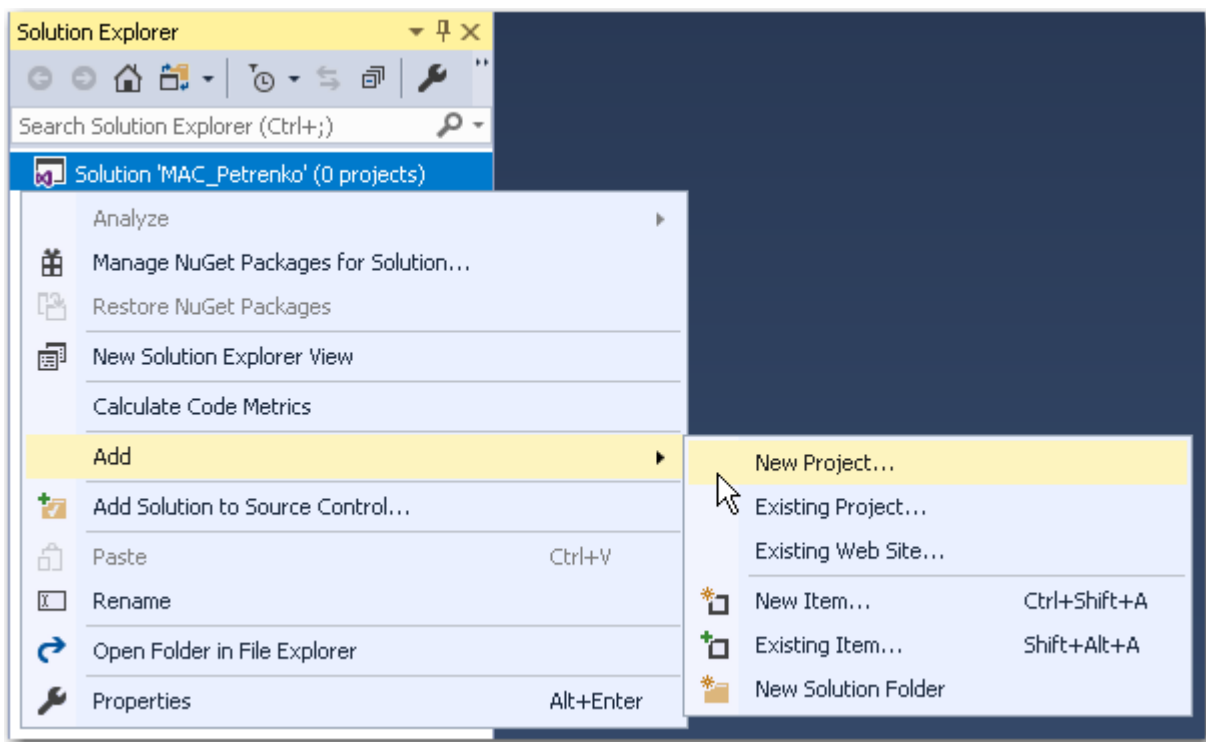
Помните, что ошибку, допущенную на этапе генерации любого из проектов **MSVS**, очень трудно впоследствии исправить (если это вообще возможно).

Имя проекта (рабочего пространства) формируется в соответствии с фамилией и инициалами студента.

В нашем примере это виртуальный студент – **Петренко Игорь Николаевич**.

Желательно располагать рабочее пространство (позиция **Location:**) в корневом каталоге одного из логических разделов вашего жесткого диска.

Теперь, в уже имеющемся рабочем пространстве **MAC\_Petrenko**, сгенерируем новый проект – проект динамической библиотеки **MAC\_DLL**:



Первые три символа в имени библиотеки – **MAC** – заглавные буквы названия данной учебной дисциплины – **Methods of Approximate Calculations**.

Библиотеки, предназначенные для других учебных дисциплин, Вы можете также снабжать подобными литерными «масками».

Имена (и типы) других проектов, последовательно включаемых в рабочее пространство **MAC\_Petrenko**, будут формироваться в соответствии с их практическим предназначением.

На этом подготовительную работу по организации программного обеспечения учебной дисциплины «*Численные Методы*» можно считать завершённой.