[TestMethod]

public void CheckDataHandlerSaveAndLoadManyPersons()

{

// #### ARRANGE ####

DataHandler handler = new DataHandler("Data.txt");

Person[] persons = new Person[]

{

new Person("William Jensen", new DateTime(1975, 8, 24), 175.9, false, 2),

new Person("Alfred Nielsen", new DateTime(1991, 3, 12), 185.0, true, 3),

new Person("Oskar Hansen", new DateTime(2005, 11, 9), 183.2, true, 1),

new Person("Emma Pedersen", new DateTime(2013, 9, 25), 113.7, false, 0),

new Person("Alma Andersen", new DateTime(1982, 1, 5), 169.9, false, 2),

new Person("Clara Christensen", new DateTime(1999, 7, 13), 165.3, true, 0),

};

// #### ACT ####

handler.SavePersons(persons);

Person[] loadedPersons = handler.LoadPersons();

// #### ASSERT ####

Assert.AreEqual("Emma Pedersen", loadedPersons[3].Name);

Assert.AreEqual(185.0, loadedPersons[1].Height);

Assert.AreEqual(1999, loadedPersons[5].BirthDate.Year);

Assert.AreEqual(false, loadedPersons[4].IsMarried);

Assert.AreEqual(1, loadedPersons[2].NoOfChildren);

Assert.AreEqual(8, loadedPersons[0].BirthDate.Month);

}

//[TestMethod]

//public void CheckPersonConstructorOverloading()

//{

// #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true);

// #### ACT ####

// string title = person.MakeTitle();

// #### ASSERT ####

// Assert.AreEqual("Anders Andersen;24-08-1975 00:00:00;175,9;True;0", title);

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonConstructorInvalidHeight()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), -10.0, true, 3);

// // #### ACT ####

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonConstructorInvalidName()

//{

// // #### ARRANGE ####

// Person person = new Person("", new DateTime(1975, 8, 24), 175.9, true, 3);

// // #### ACT ####

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonConstructorInvalidBirthDate()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1875, 8, 24), 175.9, true, 3);

// // #### ACT ####

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonConstructorInvalidNoOfChildren()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true, -3);

// // #### ACT ####

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonInvalidNameChange()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true, 3);

// // #### ACT ####

// person.Name = "";

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonInvalidBirthDateChange()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true, 3);

// // #### ACT ####

// person.BirthDate = new DateTime(1875, 8, 24);

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonInvalidHeightChange()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true, 3);

// // #### ACT ####

// person.Height = -10.0;

// // #### ASSERT ####

//}

//[TestMethod]

//[ExpectedException(typeof(Exception))]

//public void CheckPersonInvalidNoOfChildrenChange()

//{

// // #### ARRANGE ####

// Person person = new Person("Anders Andersen", new DateTime(1975, 8, 24), 175.9, true, 3);

// // #### ACT ####

// person.NoOfChildren = -3;

// // #### ASSERT ####

//}