

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ УНИВЕРСИТЕТ “СИНЕРГИЯ”»
Факультет Информационных технологий
Кафедра Цифровой экономики

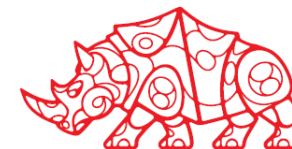
ОТЧЕТ
о прохождении производственной практики
по профессиональному модулю
ПМ.02 Осуществление интеграции программных модулей
в период с «08» июня 2025 г. по «21» июня 2025 г.
Специальность 09.02.07 Информационные системы и
программирование

ФИО обучающегося: Шишков Сергей Владимирович
Группа: ДКИП-205 ПРОГ
ФИО Руководителя: Сибирев Иван Валерьевич



Содержание

1. Инструктаж по соблюдению правил противопожарной безопасности, правил охраны труда, техники безопасности, санитарно-эпидемиологических правил и гигиенических нормативов
2. Ознакомление с инструментальными средствами
3. Сбор информации об объекте практики и анализ содержания источников
4. Экспериментально-практическая работа. Приобретение необходимых умений и практического опыта работы по специальности в рамках освоения вида деятельности ВД 2. Осуществление интеграции программных модулей
5. Обработка и систематизация полученного фактического материала



Организационный этап

Пример заполнения

Правила внутреннего распорядка, правила и нормы охраны труда, техники безопасности при работе с вычислительной техникой

Я, **Шишков Сергей Владимирович**, проходил производственную практику на базе Университета «Синергия».

При выполнении индивидуального задания по практике решал(а) кейс Packet

Создайте класс, представляющий сетевой пакет, включая его заголовок и данные. Необходимо разработать класс `NetworkPacket` в Unity, который будет моделировать сетевой пакет с заголовком (например, IP-заголовок) и полезными данными. Класс должен включать методы для управления заголовком, данными и проверки целостности пакета.

Перед началом практики:

- Принял участие в организационном собрании по практике.
- Ознакомился с комплектом шаблонов отчетной документации по практике.
- Уточнил контакты руководителя практики от Образовательной/ Профильной организации, а также правила в отношении субординации, внешнего вида, графика работы, техники безопасности:

Требования к внешнему виду: Деловой стиль

График работы: Понедельник – пятница с 9:00 до 18:00

Круг обязанностей: студент

Доступ к данным: студент

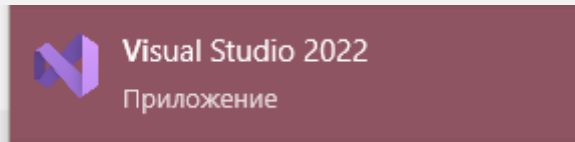


Организационный этап

Пример заполнения

Ознакомление с ПО

Ознакомиться с инструментальными средствами для выполнения производственной практики и осуществить предустановку программного обеспечения.

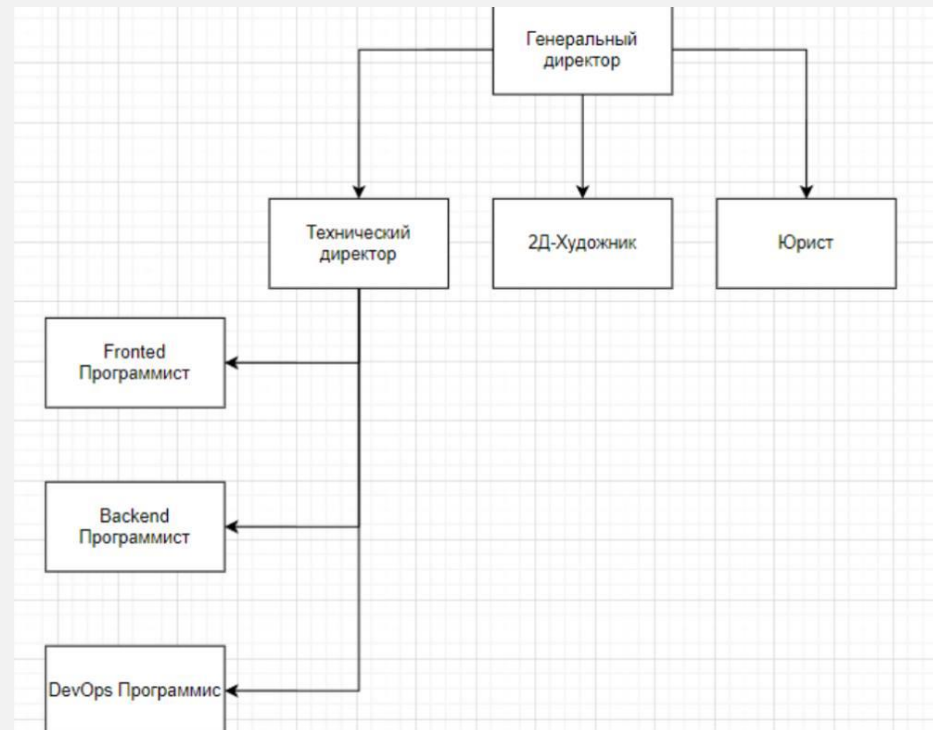


Организационный этап

Пример заполнения

Сбор информации об объекте практики и анализ содержания источников

Спроектировать организационную структуру и описать выбранную предметную область.



Этап проектирования

Пример заполнения

Интеграция модулей в программное обеспечение

```
Assembly-CSharp OSPF.Simulation.UI.NetworkUI
1  using UnityEngine;
2  using UnityEngine.UI;
3  using TMPro;
4
5  namespace OSPF.Simulation.UI
6  {
7      // Скрипт Unity | Ссылка: 0
8      public class NetworkUI : MonoBehaviour
9      {
10         public Network network;
11         public TMP_InputField routerIdInput;
12         public TMP_InputField costInput;
13         public TMP_Text routingTableText;
14
15         // Ссылка: 0
16         public void OnAddRouterClicked()
17         {
18             if (string.IsNullOrEmpty(routerIdInput.text))
19             {
20                 Debug.LogError("Router ID cannot be empty!");
21                 return;
22             }
23
24             GameObject newRouter = new GameObject("Router_" + routerIdInput.text);
25             Router routerComp = newRouter.AddComponent<Router>();
26             routerComp.routerID = routerIdInput.text;
27             network.AddRouter(routerComp);
28
29             routerIdInput.text = "";
30
31         // Ссылка: 1
32         public void UpdateRoutingTableDisplay()
33         {
34             string tableText = "Routing Tables:\n";
35             foreach (var router in network.routers.Values)
36             {
37                 tableText += $"{router.routerID}:\n";
38                 foreach (var entry in router.routingTable)
39                 {
40                     tableText += $"  -> {entry.Key} (via {entry.Value.nextHop}, cost: {entry.Value.cost})\n";
41                 }
42             }
43             routingTableText.text = tableText;
44         }
45
46         // Сообщение Unity | Ссылка: 0
47         void Update()
48         {
49             if (network.routers.Count > 0)
50             {
51                 UpdateRoutingTableDisplay();
52             }
53         }
54     }
55 }
```



Этап проектирования

Отладка программных модулей

На данном слайде необходимо продемонстрировать итоговый скриншот отладки разработанных модулей.

```
17 var header = new Dictionary<string, object>
18 {
19     { "SourceIP", "192.168.1.1" },
20     { "DestinationIP", "192.168.1.2" },
21     { "Protocol", "TCP" },
22     { "TTL", 64 }
23 };
24
25 byte[] data = System.Text.Encoding.UTF8.GetBytes("test packet data");
26 var packet = new NetworkPacket(header, data);
27
28 // Debug 1: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
29 packet.GetHeader(0) = 0x01; // Запись в пакет
30
31 // Проверка заголовка - Debug 2: Итерация по массиву
32 foreach (var field in packet.GetHeader())
33 {
34     Debug.Log($"field key: {field.Key} field value: {field.Value}");
35 }
36
37 // Debug 3: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
38 Debug.Log($"Data: {System.Text.Encoding.ASCII.GetString(packet.GetData())}");
39
40 void TestJsonSerialization()
41 {
42     Debug.Log("==== Testing JSON Serialization =====");
43
44     // Debug 4: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
45     var header = new Dictionary<string, object>
46     {
47         { "Test", true },
48         { "RandomValue", 42 },
49         { "Timestamp", System.DateTime.UtcNow },
50         { "HeaderData", new Dictionary<string, object> { } }
51     };
52
53     byte[] data = new byte[100];
54     new System.Random().NextBytes(data);
55
56     var originalPacket = new NetworkPacket(header, data);
57
58     // Debug 5: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
59     string json = originalPacket.ToJson();
60     Debug.Log($"Serialized JSON: {json}");
61
62     // Debug 6: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
63     var deserializedPacket = NetworkPacket.FromJson(json);
64     Debug.Log($"Deserialized checksum valid: {deserializedPacket.VerifyChecksum()}");
65
66     void TestChecksumVerification()
67     {
68         Debug.Log("==== Testing Checksum Verification =====");
69
70         // Debug 7: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
71         var packet = new NetworkPacket(
72             new Dictionary<string, object> { { "Test", "Checksum" } },
73             null
74         );
75
76         // Debug 8: Проверка структуры пакета (по шаблону из-за записи в GetHeader)
77         Debug.Log($"Original checksum valid: {packet.VerifyChecksum()}");
78
79         // Проверка структуры пакета (по шаблону из-за записи в GetHeader)
80         var corruptedData = packet.GetData();
81         if (corruptedData != null && corruptedData.Length > 0) // Это исправлено, но в коде его нет
82         {
83             corruptedData[0] = 0xFF;
84         }
85
86         var corruptedPacket = new NetworkPacket(
87             packet.Header,
88             corruptedData
89         );
90     }
91 }
```

```
[20:43:14] Assets/NetworkPacketTest.cs(87,42): error CS1526: A new expression requires an argument list or (), [], or {} after type
[20:43:14] Assets/NetworkPacketTest.cs(87,42): error CS1002: ; expected
[20:43:14] Assets/NetworkPacketTest.cs(87,42): error CS1513: } expected
```

Пример заполнения

```
19 {
20     { "SourceIP", "192.168.1.1" },
21     { "DestinationIP", "192.168.1.2" },
22     { "Protocol", "TCP" },
23     { "TTL", 64 }
24 };
25
26 byte[] data = System.Text.Encoding.UTF8.GetBytes("test packet data");
27 var packet = new NetworkPacket(header, data);
28
29 // Проверка структуры пакета (по шаблону из-за записи в GetHeader)
30 foreach (var field in packet.GetHeader())
31 {
32     Debug.Log($"field key: {field.Key} field value: {field.Value}");
33 }
34
35 // Проверка структуры пакета (по шаблону из-за записи в GetHeader)
36 Debug.Log($"Data: {System.Text.Encoding.UTF8.GetString(packet.GetData())}");
37
38 // Проверка структуры пакета (по шаблону из-за записи в GetHeader)
39 Debug.Log($"Checksum valid: {packet.VerifyChecksum()}");
40
41 void TestJsonSerialization()
42 {
43     Debug.Log("==== Testing JSON Serialization =====");
44
45     var header = new Dictionary<string, object>
46     {
47         { "Test", true },
48         { "RandomValue", 42 },
49         { "Timestamp", System.DateTime.UtcNow },
50         { "HeaderData", new Dictionary<string, object> { } }
51     };
52
53     byte[] data = new byte[100];
54     new System.Random().NextBytes(data);
55
56     var originalPacket = new NetworkPacket(header, data);
57     string json = originalPacket.ToJson();
58     Debug.Log($"Serialized JSON: {json}");
59
60     var deserializedPacket = NetworkPacket.FromJson(json);
61     Debug.Log($"Deserialized checksum valid: {deserializedPacket.VerifyChecksum()}");
62
63     void TestChecksumVerification()
64     {
65         Debug.Log("==== Testing Checksum Verification =====");
66
67         var packet = new NetworkPacket(
68             new Dictionary<string, object> { { "Test", "Checksum" } },
69             System.Text.Encoding.UTF8.GetBytes("test data")
70         );
71
72         Debug.Log($"Original checksum valid: {packet.VerifyChecksum()}");
73
74         // Проверка структуры пакета (по шаблону из-за записи в GetHeader)
75         var corruptedData = packet.GetData();
76         corruptedData[0] = 0xFF; // Исправлено, но в коде его нет
77         var corruptedPacket = new NetworkPacket(packet.Header, corruptedData);
78         Debug.Log($"Corrupted checksum valid: {corruptedPacket.VerifyChecksum()}");
79     }
80 }
```

Проблемы не найдены. | 0 Ошибки | 0 Предупреждения | 0 из 4 Сообщения | Сборка и IntelliSense



Формирование отчетной документации по результатам работ

При оформлении отчетных материалов следует придерживаться действующих стандартов.

- В соответствии с ГОСТ 2.105-79 «Общие требования к текстовым документам» иллюстрации (графики, схемы, диаграммы) могут быть приведены как в основном тексте, так и в приложении. Все иллюстрации именуют рисунками. Все рисунки, таблицы и формулы нумеруют арабскими цифрами последовательно (сквозная нумерация) или в пределах раздела (относительная нумерация). В приложении - в пределах приложения. Каждый рисунок должен иметь подрисуночную подпись - название, помещаемую под рисунком.
- Рисунки следует размещать так, чтобы их можно было рассматривать без поворота страницы. Если такое размещение невозможно, рисунки следует располагать так, чтобы для просмотра надо было повернуть страницу по часовой стрелке. В этом случае верхним краем является левый край страницы. Расположение и размеры полей сохраняются.
- Номер таблицы размещают в правом верхнем углу или перед заголовком таблицы, если он есть. Заголовок, кроме первой буквы, выполняют строчными буквами. Ссылки на таблицы в тексте пояснительной записки указывают в виде слова «табл.» и номера таблицы. *Например: Результаты тестов приведены в табл. 4.*



Формирование отчетной документации по результатам работ

При оформлении отчетных материалов следует придерживаться действующих стандартов.

- Список литературы должен включать все использованные источники. Сведения о книгах (монографиях, учебниках, пособиях, справочниках и т.д.) должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство, год издания. При наличии трех и более авторов допускается указывать фамилию и инициалы только первого из них со словами «и др.». Издательство надо приводить полностью в именительном падеже: допускается сокращение названия только двух городов: Москва (М.) и Санкт-Петербург (СПб.).
- Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, наименование статьи, издания (журнала), серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и номера страниц, на которых помещена статья.
- При ссылке на источник из списка литературы (особенно при обзоре аналогов) надо указывать порядковый номер по списку литературы, заключенный в квадратные скобки; например: [5].



Выводы о результатах прохождения производственной практики:
выполняемая работа, приобретенные умения и навыки

Подведите итоги прохождения производственной практики:

В ходе прохождения производственной практики мной были освоены следующие навыки:

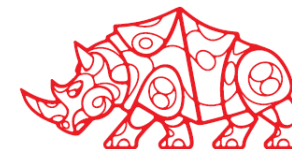
1. Программирование на C#
2. Работа в Unity
3. Отладка и тестирование
4. Работа с технической документацией
5. разработать класс NetworkPacket в Unity



Отчетный этап

Список используемой литературы

1. Советов, Б. Я. Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 420 с. — (Профессиональное образование). — ISBN 978-5-534-09324-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/514585>
2. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2023. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518499>
3. Нагаева, И.А. Основы алгоритмизации и программирования: практикум : [12+] / И.А. Нагаева, И.А. Кузнецов. — Москва ; Берлин : Директ-Медиа, 2021. — 169 с. : схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/>
4. ...
5. ...



Отчетный этап

Приложения

Пример заполнения

Приложение 1

Приложение 2

