

## Лабораторна робота № 1

**Тема:** Принципи програмування. DRY, KISS, SOLID, YAGNI та ін.

**Мета роботи:** навчитися дотримуватися принципів програмування та обґрунтовувати їх.

### Хід роботи

**Завдання №1-2.** Виконати завдання з дотриманням відомих Вам принципів програмування.

1. Створіть систему класів для обліку зоопарку. Ви можете створювати класи для різних видів і підвидів тварин; для вольєрів різних розмірів і типів; корму для тварин; працівників зоопарку.

2. Створіть класи інвентаризації, для виведення на екран інформації про наявних тварин, кількості співробітників тощо.

#### Лістинг програми:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ZooManagementSystem
{
    public class Animal
    {
        public string Species { get; set; }
        public string Subspecies { get; set; }
        public int Age { get; set; }
    }
    public class Enclosure
    {
        public int Id { get; set; }
        public string Type { get; set; }
        public double Area { get; set; }
        public List<Animal> Animals { get; set; } = new List<Animal>();
    }
    public class Employee
    {
        public string Name { get; set; }
        public int Age { get; set; }
        public string Position { get; set; }
    }
    public class Food
    {
        public string Name { get; set; }
        public int Quantity { get; set; }
    }
    public class Zoo
    {
        public List<Food> Foods { get; set; }
        public List<Animal> Animals { get; set; }
        public List<Enclosure> Enclosures { get; set; }
        public List<Employee> Employees { get; set; }

        public Zoo()
        {
            Foods = new List<Food>();
            Animals = new List<Animal>();
            Enclosures = new List<Enclosure>();
            Employees = new List<Employee>();
        }
    }
}
```

		Foods = new List<Food>();			ДУ «Житомирська політехніка».24.121.17.000–Лр1				
		Animals = new List<Animal>();							
		Enclosures = new List<Enclosure>();							
Змн.	Арк.	Employee = new List<Employee>();	Прим.	Дата	Звіт з лабораторної роботи				
Розроб.		Леус В.О.							
Перевір.		Фант М.О.							
Керівник									
Н. контр.									
Зав. каф.									
					Лім.	Арк.	Аркушів		
						1	9		
					ФІКТ Гр. ІПЗ-22-3				

```

class Program
{
    static void Main(string[] args)
    {
        Console.OutputEncoding = Encoding.UTF8;

        Zoo zoo = new Zoo();
        InitializeZoo(zoo);

        while (true)
        {
            Console.Clear();
            Console.WriteLine("Choose an action:");
            Console.WriteLine("1 - Action with animals");
            Console.WriteLine("2 - Action with enclosures");
            Console.WriteLine("3 - Action with employees");
            Console.WriteLine("4 - Action with food");
            Console.WriteLine("5 - View All");
            Console.WriteLine("Exit");

            string choice = Console.ReadLine();

            switch (choice)
            {
                case "1":
                    AnimalActions(zoo);
                    break;
                case "2":
                    EnclosureActions(zoo);
                    break;
                case "3":
                    EmployeeActions(zoo);
                    break;
                case "4":
                    FoodActions(zoo);
                    break;
                case "5":
                    ViewAll(zoo);
                    break;
                case "Exit":
                case "exit":
                    Environment.Exit(0);
                    break;
                default:
                    Console.WriteLine("Invalid selection. Please try again.");
                    break;
            }
        }
    }

    static void InitializeZoo(Zoo zoo)
    {
        zoo.Animals.Add(new Animal { Species = "Lion", Subspecies = "African", Age = 5 });
        zoo.Animals.Add(new Animal { Species = "Tiger", Subspecies = "Bengal", Age = 4 });
        zoo.Animals.Add(new Animal { Species = "Elephant", Subspecies = "Indian", Age = 10 });

        zoo.Enclosures.Add(new Enclosure { Id = 1, Type = "Medium 1", Area = 100 });
        zoo.Enclosures.Add(new Enclosure { Id = 2, Type = "Medium 2", Area = 80 });
        zoo.Enclosures.Add(new Enclosure { Id = 3, Type = "Large 1", Area = 200 });

        zoo.Employees.Add(new Employee { Name = "Ivan", Age = 35, Position = "Veterinarian" });
        zoo.Employees.Add(new Employee { Name = "Maria", Age = 28, Position = "Maintenance" });
        zoo.Employees.Add(new Employee { Name = "Petro", Age = 40, Position = "Guardian" });

        zoo.Foods.Add(new Food { Name = "Meat", Quantity = 100 });
        zoo.Foods.Add(new Food { Name = "Vegetables", Quantity = 200 });
        zoo.Foods.Add(new Food { Name = "Fruit", Quantity = 150 });
    }

    static void AnimalActions(Zoo zoo)
    {
        Console.OutputEncoding = Encoding.UTF8;
        int newAnimalAge;

        while (true)
        {
            Console.Clear();

```

		Левс В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Пр1	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

Console.WriteLine(" Action with animals:");
Console.WriteLine("1 - View all animals");
Console.WriteLine("2 - Change animal data");
Console.WriteLine("3 - Delete animal");
Console.WriteLine("4 - Add an animal");
Console.WriteLine("5 - Return to main menu");

string choice = Console.ReadLine();

switch (choice)
{
    case "1":
        Console.WriteLine("List of all animals:");
        foreach (var animal in zoo.Animals)
        {
            Console.WriteLine($"Species: {animal.Species}, Subspecies: {animal.Subspecies}, Age: {animal.Age}");
        }
        break;
    case "2":
        Console.WriteLine("Enter the type of animal to change data:");
        string speciesToEdit = Console.ReadLine();
        Console.WriteLine("Enter the animal subspecies to change data:");
        string subspeciesToEdit = Console.ReadLine();
        Animal animalToEdit = zoo.Animals.Find(a => a.Species == speciesToEdit && a.Subspecies == subspeciesToEdit);
        if (animalToEdit != null)
        {
            Console.WriteLine($"Enter new age for animal {speciesToEdit} {subspeciesToEdit}:");
            if (int.TryParse(Console.ReadLine(), out newAnimalAge))
            {
                animalToEdit.Age = newAnimalAge;
                Console.WriteLine("Data changed successfully.");
            }
            else
            {
                Console.WriteLine("Invalid age format. Please enter an integer.");
            }
        }
        else
        {
            Console.WriteLine("No animal found.");
        }
        break;
    case "3":
        Console.WriteLine("Enter the type of animal to delete:");
        string speciesToDelete = Console.ReadLine();
        Console.WriteLine("Enter the animal subspecies to delete:");
        string subspeciesToDelete = Console.ReadLine();
        Animal animalToDelete = zoo.Animals.Find(a => a.Species == speciesToDelete && a.Subspecies == subspeciesToDelete);
        if (animalToDelete != null)
        {
            zoo.Animals.Remove(animalToDelete);
            Console.WriteLine($"Animal {speciesToDelete} {subspeciesToDelete} has been removed from the zoo.");
        }
        else
        {
            Console.WriteLine("No animal found.");
        }
        break;
    case "4":
        Console.WriteLine("Enter the appearance of the new animal:");
        string newSpecies = Console.ReadLine();
        Console.WriteLine("Enter the subspecies of the new animal:");
        string newSubspecies = Console.ReadLine();
        Console.WriteLine("Enter the age of the new animal:");
        int newAge = int.Parse(Console.ReadLine());
        zoo.Animals.Add(new Animal { Species = newSpecies, Subspecies = newSubspecies, Age = newAge });
        Console.WriteLine($"Animal {newSpecies} {newSubspecies} has been added to the zoo.");
        break;
    case "5":
        return;
    default:
        Console.WriteLine("Invalid selection. Please try again.");
        break;
}
Console.WriteLine("Press Enter to continue...");
Console.ReadLine();
}
}

```

		Левс В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Лр1	Арк.
		Фант М.О.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

static void EnclosureActions(Zoo zoo)
{
    Console.OutputEncoding = Encoding.UTF8;
    int newEnclosureId;
    while (true)
    {
        Console.Clear();
        Console.WriteLine("Action with enclosures:");
        Console.WriteLine("1 - View all enclosures");
        Console.WriteLine("2 - Change data about the enclosure");
        Console.WriteLine("3 - Delete enclosure");
        Console.WriteLine("4 - Add enclosure");
        Console.WriteLine("5 - Return to main menu");

        string choice = Console.ReadLine();

        switch (choice)
        {
            case "1":
                Console.WriteLine("List of all enclosures:");
                foreach (var enclosure in zoo.Enclosures)
                {
                    Console.WriteLine($"Id: {enclosure.Id}, Type: {enclosure.Type}, Area: {enclosure.Area}");
                }
                break;
            case "2":
                Console.WriteLine("Enter enclosure ID to change data:");
                int enclosureIdToEdit = int.Parse(Console.ReadLine());
                Enclosure enclosureToEdit = zoo.Enclosures.Find(e => e.Id == enclosureIdToEdit);
                if (enclosureToEdit != null)
                {
                    Console.WriteLine($"Enter a new type for enclosure with ID {enclosureIdToEdit}:");
                    string newType = Console.ReadLine();
                    Console.WriteLine($"Enter a new area for the enclosure with the ID {enclosureIdToEdit}:");
                    double newArea = double.Parse(Console.ReadLine());
                    enclosureToEdit.Type = newType;
                    enclosureToEdit.Area = newArea;
                    Console.WriteLine("Data changed successfully.");
                }
                else
                {
                    Console.WriteLine("Aviary not found");
                }
                break;
            case "3":
                Console.WriteLine("Enter the enclosure ID to delete:");
                int enclosureIdToDelete = int.Parse(Console.ReadLine());
                Enclosure enclosureToDelete = zoo.Enclosures.Find(e => e.Id == enclosureIdToDelete);
                if (enclosureToDelete != null)
                {
                    zoo.Enclosures.Remove(enclosureToDelete);
                    Console.WriteLine($"The enclosure with ID {enclosureIdToDelete} has been deleted from the zoo.");
                }
                else
                {
                    Console.WriteLine("Aviary not found");
                }
                break;
            case "4":
                Console.WriteLine("Enter the ID of the new enclosure:");
                newEnclosureId = int.Parse(Console.ReadLine());
                Console.WriteLine("Enter the type of new enclosure:");
                string newEnclosureType = Console.ReadLine();
                Console.WriteLine("Enter the area of the new enclosure:");
                double newEnclosureArea = double.Parse(Console.ReadLine());
                zoo.Enclosures.Add(new Enclosure { Id = newEnclosureId, Type = newEnclosureType, Area = newEnclosureArea });
                Console.WriteLine($"The enclosure with ID {newEnclosureId} has been added to the zoo.");
                break;
            case "5":
                return;
            default:
                Console.WriteLine("Invalid selection. Please try again.");
                break;
        }

        Console.WriteLine("Press Enter to continue...");
        Console.ReadLine();
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Лр1	Арк.
		Фант М.О.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}

static void EmployeeActions(Zoo zoo)
{
    Console.OutputEncoding = Encoding.UTF8;
    while (true)
    {
        Console.Clear();
        Console.WriteLine("Action with employees:");
        Console.WriteLine("1 - View all employees");
        Console.WriteLine("2 - Change employee data");
        Console.WriteLine("3 - Delete employee");
        Console.WriteLine("4 - Add an employee");
        Console.WriteLine("5 - Return to main menu");

        string choice = Console.ReadLine();

        switch (choice)
        {
            case "1":
                Console.WriteLine("List of all workers:");
                foreach (var employee in zoo.Employees)
                {
                    Console.WriteLine($"Name: {employee.Name}, Age: {employee.Age}, Position: {employee.Position}");
                }
                break;
            case "2":
                Console.WriteLine("Enter the name of the employee to change the data:");
                string nameToEdit = Console.ReadLine();
                Employee employeeToEdit = zoo.Employees.Find(e => e.Name == nameToEdit);
                if (employeeToEdit != null)
                {
                    Console.WriteLine($"Enter new age for employee {nameToEdit}:");
                    if (int.TryParse(Console.ReadLine(), out int newAge))
                    {
                        employeeToEdit.Age = newAge;
                        Console.WriteLine("Data changed successfully.");
                    }
                    else
                    {
                        Console.WriteLine("Invalid age format. Please enter an integer.");
                    }
                }
                else
                {
                    Console.WriteLine("Employee not found.");
                }
                break;
            case "3":
                Console.WriteLine("Enter the name of the employee to delete:");
                string nameToDelete = Console.ReadLine();
                Employee employeeToDelete = zoo.Employees.Find(e => e.Name == nameToDelete);
                if (employeeToDelete != null)
                {
                    zoo.Employees.Remove(employeeToDelete);
                    Console.WriteLine($"Employee {nameToDelete} has been removed from the zoo.");
                }
                else
                {
                    Console.WriteLine("Employee not found.");
                }
                break;
            case "4":
                Console.WriteLine("Enter the name of the new employee:");
                string newName = Console.ReadLine();
                Console.WriteLine($"Enter age for employee {newName}:");
                int newEmployeeAge;
                if (int.TryParse(Console.ReadLine(), out newEmployeeAge))
                {
                    Console.WriteLine($"Enter the job title for employee {newName}:");
                    string newPosition = Console.ReadLine();
                    zoo.Employees.Add(new Employee { Name = newName, Age = newEmployeeAge, Position = newPosition });
                    Console.WriteLine($"Employee {newName} has been added to the zoo.");
                }
                else
                {
                    Console.WriteLine("Invalid age format. Please enter an integer.");
                }
            }
        }
    }
}

```

		Левс В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Пр1	Арк.
		Фант М.О.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        }
        break;
    case "5":
        return;
    default:
        Console.WriteLine("Invalid selection. Please try again.");
        break;
    }

    Console.WriteLine("Press Enter to continue...");
    Console.ReadLine();
}

static void FoodActions(Zoo zoo)
{
    Console.OutputEncoding = Encoding.UTF8;
    while (true)
    {
        Console.Clear();
        Console.WriteLine("Action with food:");
        Console.WriteLine("1 - view the list of available food");
        Console.WriteLine("2 - add a new type of food");
        Console.WriteLine("3 - remove type of food");
        Console.WriteLine("4 - Return to main menu");

        string choice = Console.ReadLine();

        switch (choice)
        {
            case "1":
                Console.WriteLine("List of available food:");
                foreach (var food in zoo.Foods)
                {
                    Console.WriteLine($"Name: {food.Name}, Quantity: {food.Quantity}");
                }
                break;
            case "2":
                Console.WriteLine("Enter the name of the new type of food:");
                string newFoodName = Console.ReadLine();
                Console.WriteLine("Enter the amount of food:");
                int newFoodQuantity = int.Parse(Console.ReadLine());
                zoo.Foods.Add(new Food { Name = newFoodName, Quantity = newFoodQuantity });
                Console.WriteLine($"Food type {newFoodName} added.");
                break;
            case "3":
                Console.WriteLine("Enter the name of the type of food to delete:");
                string foodToDelete = Console.ReadLine();
                Food foodItemToDelete = zoo.Foods.Find(f => f.Name == foodToDelete);
                if (foodItemToDelete != null)
                {
                    zoo.Foods.Remove(foodItemToDelete);
                    Console.WriteLine($"Food view {foodToDelete} deleted.");
                }
                else
                {
                    Console.WriteLine($"Food type {foodToDelete} not found.");
                }
                break;
            case "4":
                return;
            default:
                Console.WriteLine("Invalid selection. Please try again.");
                break;
        }

        Console.WriteLine("Press Enter to continue...");
        Console.ReadLine();
    }
}

static void ViewAll(Zoo zoo)
{
    Console.OutputEncoding = Encoding.UTF8;
    Console.Clear();
    Console.WriteLine("Zoo Information:");
    Console.WriteLine("Animals:");
    foreach (var animal in zoo.Animals)
    {

```

		Левс В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Лр1	Арк.
		Фант М.О.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        Console.WriteLine($"Species: {animal.Species}, Subspecies: {animal.Subspecies}, Age: {animal.Age}");
    }
    Console.WriteLine("\nAviaries:");
    foreach (var enclosure in zoo.Enclosures)
    {
        Console.WriteLine($"Id: {enclosure.Id}, Type: {enclosure.Type}, Area: {enclosure.Area}");
    }
    Console.WriteLine("\nEmployees:");
    foreach (var employee in zoo.Employees)
    {
        Console.WriteLine($"Name: {employee.Name}, Age: {employee.Age}, Position: {employee.Position}");
    }
    Console.WriteLine("\nPress Enter to continue...");
    Console.ReadLine();
}
}
}

```

```

Choose an action:
1 - Action with animals
2 - Action with enclosures
3 - Action with employees
4 - Action with food
5 - View All
Exit

```

Мал.1. Головне меню

```

Action with animals:
1 - View all animals
2 - Change animal data
3 - Delete animal
4 - Add an animal
5 - Return to main menu
1
List of all animals:
Species: Lion, Subspecies: African, Age: 5
Species: Tiger, Subspecies: Bengal, Age: 4
Species: Elephant, Subspecies: Indian, Age: 10
Press Enter to continue...

```

Мал.2. Перегляд тварин

```

Action with animals:
1 - View all animals
2 - Change animal data
3 - Delete animal
4 - Add an animal
5 - Return to main menu
4
Enter the appearance of the new animal:
Rhino
Enter the subspecies of the new animal:
White
Enter the age of the new animal:
12
Animal Rhino White has been added to the zoo.
Press Enter to continue...

```

Мал.3. Додавання нової тварини

За аналогією до тварин зроблені меню для вольєрів, працівників та продуктів.

		Леус В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Пр1	Арк.
		Фант М.О.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

Zoo Information:
Animals:
Species: Lion, Subspecies: African, Age: 5
Species: Tiger, Subspecies: Bengal, Age: 4
Species: Elephant, Subspecies: Indian, Age: 10
Species: Rhino, Subspecies: White, Age: 12

Aviaries:
Id: 1, Type: Medium 1, Area: 100
Id: 2, Type: Medium 2, Area: 80
Id: 3, Type: Large 1, Area: 200

Employees:
Name: Ivan, Age: 35, Position: Veterinarian
Name: Maria, Age: 28, Position: Maintenance
Name: Petro, Age: 40, Position: Guardian

Press Enter to continue...

```

#### Мал.4. Перегляд інвертизації

**Завдання № 3:** Опишіть особливості дотримання принципів програмування в Вашому коді

1. Додайте файл README.md в кореневу директорію цієї лабораторної роботи. В файлі README.md опишіть дотримання окремо кожного принципу програмування, який Вам відомо, і який можна продемонструвати Вашим кодом.

2. Опис можна залишати українською або (бажано) англійською мовами.

3. Опис повинен містити посилання на відповідні файли і рядки коду.

4. Для отримання максимальної оцінки потрібно продемонструвати мінімум 7 принципів. SOLID принципи рахуються окремо. Повний список принципів, які було розглянуто на лекції:

- a. DRY,
- b. KISS,
- c. SOLID (5 окремих принципів)
- d. YAGNI
- e. Composition Over Inheritance
- f. Program to Interfaces not Implementations
- g. Fail Fast

		Леус В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Лр1	Арк.
		Фант М.О.				8
Змн.	Арк.	№ докум.	Підпис	Дата		



```

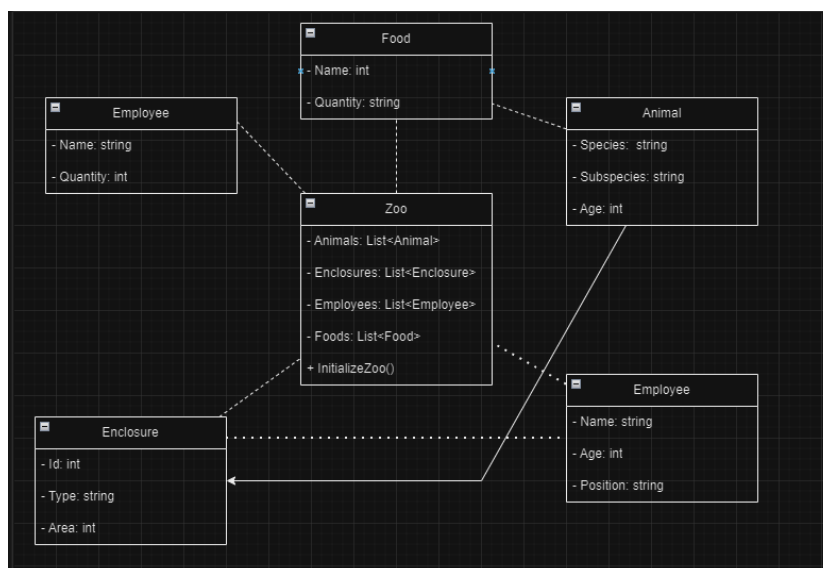
1  README.md
2  README.md
3  1  **B. KISS (Keep It Simple, Stupid):**
4  2  The KISS principle implies that the system should be as simple and understandable as possible. In this code, we try to keep the logic simple, using simple data
5  3  structures and clear methods. For example, the ViewAll method simply prints information about the zoo to the console making the code clear and easy to understand.
6  4  You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L31).
7  5  **Single Responsibility Principle:** Each class is responsible for only one piece of functionality. For example, the Zoo, Animal, Enclosure, Employee, and Food
8  6  classes are responsible for managing their respective zoo entities.
9  7  You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L31).
10 8  **Open/Closed Principle:** Classes are written to be open to extension but closed to modification. For example, you can easily add new methods or properties to
11 9  classes without changing existing code.
12 10 You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L31).
13 11 **Liskov Substitution Principle:** Classes can be replaced by instances of their subtypes without affecting the correctness of the program. For example, if we have
14 12 a method that takes an object of type Animal, we can pass in any class that inherits from Animal, such as Lion or Tiger.
15 13 You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L182).
16 14 **D. YAGNI (You Aren't Gonna Need It):** The YAGNI principle suggests not adding functionality that is not currently needed. There is no obvious violation of this
17 15 principle in the code, since functionality is added only when necessary (for example, methods for adding, removing, and viewing elements).
18 16 You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L372).
19 17 **E. Composition Over Inheritance:** The principle is that it is better to use object composition than class inheritance. There are no examples of inheritance in
20 18 the code above, but there are examples of object composition, for example, Zoo contains the lists Animals, Enclosures, Employees, and Foods.
21 19 You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L31).
22 20 **F. Program to Interfaces not Implementations:** The code uses an interface-based approach, which allows for more flexible dependency management and easier
23 21 testing. For example, methods for working with Zoo objects accept the Zoo interface instead of a specific implementation, making it easy to replace implementations
24 22 as needed.
25 23 You can marvel at the butt here [Program.cs](KPZ-LAB-01/KPZ-LAB-01/Program.cs#L116).
26 24
27 25
28 26
29 27
30 28
31 29
32 30
33 31
34 32
35 33
36 34
37 35
38 36
39 37
40 38
41 39
42 40
43 41
44 42
45 43
46 44
47 45
48 46
49 47
50 48
51 49
52 50
53 51
54 52
55 53
56 54
57 55
58 56
59 57
60 58
61 59
62 60
63 61
64 62
65 63
66 64
67 65
68 66
69 67
70 68
71 69
72 70
73 71
74 72
75 73
76 74
77 75
78 76
79 77
80 78
81 79
82 80
83 81
84 82
85 83
86 84
87 85
88 86
89 87
90 88
91 89
92 90
93 91
94 92
95 93
96 94
97 95
98 96
99 97
100 98
101 99
102 100
103 101
104 102
105 103
106 104
107 105
108 106
109 107
110 108
111 109
112 110
113 111
114 112
115 113
116 114
117 115
118 116
119 117
120 118
121 119
122 120
123 121
124 122
125 123
126 124
127 125
128 126
129 127
130 128
131 129
132 130
133 131
134 132
135 133
136 134
137 135
138 136
139 137
140 138
141 139
142 140
143 141
144 142
145 143
146 144
147 145
148 146
149 147
150 148
151 149
152 150
153 151
154 152
155 153
156 154
157 155
158 156
159 157
160 158
161 159
162 160
163 161
164 162
165 163
166 164
167 165
168 166
169 167
170 168
171 169
172 170
173 171
174 172
175 173
176 174
177 175
178 176
179 177
180 178
181 179
182 180
183 181
184 182
185 183
186 184
187 185
188 186
189 187
190 188
191 189
192 190
193 191
194 192
195 193
196 194
197 195
198 196
199 197
200 198
201 199
202 200
203 201
204 202
205 203
206 204
207 205
208 206
209 207
210 208
211 209
212 210
213 211
214 212
215 213
216 214
217 215
218 216
219 217
220 218
221 219
222 220
223 221
224 222
225 223
226 224
227 225
228 226
229 227
230 228
231 229
232 230
233 231
234 232
235 233
236 234
237 235
238 236
239 237
240 238
241 239
242 240
243 241
244 242
245 243
246 244
247 245
248 246
249 247
250 248
251 249
252 250
253 251
254 252
255 253
256 254
257 255
258 256
259 257
260 258
261 259
262 260
263 261
264 262
265 263
266 264
267 265
268 266
269 267
270 268
271 269
272 270
273 271
274 272
275 273
276 274
277 275
278 276
279 277
280 278
281 279
282 280
283 281
284 282
285 283
286 284
287 285
288 286
289 287
290 288
291 289
292 290
293 291
294 292
295 293
296 294
297 295
298 296
299 297
300 298
301 299
302 300
303 301
304 302
305 303
306 304
307 305
308 306
309 307
310 308
311 309
312 310
313 311
314 312
315 313
316 314
317 315
318 316
319 317
320 318
321 319
322 320
323 321
324 322
325 323
326 324
327 325
328 326
329 327
330 328
331 329
332 330
333 331
334 332
335 333
336 334
337 335
338 336
339 337
340 338
341 339
342 340
343 341
344 342
345 343
346 344
347 345
348 346
349 347
350 348
351 349
352 350
353 351
354 352
355 353
356 354
357 355
358 356
359 357
360 358
361 359
362 360
363 361
364 362
365 363
366 364
367 365
368 366
369 367
370 368
371 369
372 370
373 371
374 372
375 373
376 374
377 375
378 376
379 377
380 378
381 379
382 380
383 381
384 382
385 383
386 384
387 385
388 386
389 387
390 388
391 389
392 390
393 391
394 392
395 393
396 394
397 395
398 396
399 397
400 398
401 399
402 400
403 401
404 402
405 403
406 404
407 405
408 406
409 407
410 408
411 409
412 410
413 411
414 412
415 413
416 414
417 415
418 416
419 417
420 418
421 419
422 420
423 421
424 422
425 423
426 424
427 425
428 426
429 427
430 428
431 429
432 430
433 431
434 432
435 433
436 434
437 435
438 436
439 437
440 438
441 439
442 440
443 441
444 442
445 443
446 444
447 445
448 446
449 447
450 448
451 449
452 450
453 451
454 452
455 453
456 454
457 455
458 456
459 457
460 458
461 459
462 460
463 461
464 462
465 463
466 464
467 465
468 466
469 467
470 468
471 469
472 470
473 471
474 472
475 473
476 474
477 475
478 476
479 477
480 478
481 479
482 480
483 481
484 482
485 483
486 484
487 485
488 486
489 487
490 488
491 489
492 490
493 491
494 492
495 493
496 494
497 495
498 496
499 497
500 498
501 499
502 500
503 501
504 502
505 503
506 504
507 505
508 506
509 507
510 508
511 509
512 510
513 511
514 512
515 513
516 514
517 515
518 516
519 517
520 518
521 519
522 520
523 521
524 522
525 523
526 524
527 525
528 526
529 527
530 528
531 529
532 530
533 531
534 532
535 533
536 534
537 535
538 536
539 537
540 538
541 539
542 540
543 541
544 542
545 543
546 544
547 545
548 546
549 547
550 548
551 549
552 550
553 551
554 552
555 553
556 554
557 555
558 556
559 557
560 558
561 559
562 560
563 561
564 562
565 563
566 564
567 565
568 566
569 567
570 568
571 569
572 570
573 571
574 572
575 573
576 574
577 575
578 576
579 577
580 578
581 579
582 580
583 581
584 582
585 583
586 584
587 585
588 586
589 587
590 588
591 589
592 590
593 591
594 592
595 593
596 594
597 595
598 596
599 597
600 598
601 599
602 600
603 601
604 602
605 603
606 604
607 605
608 606
609 607
610 608
611 609
612 610
613 611
614 612
615 613
616 614
617 615
618 616
619 617
620 618
621 619
622 620
623 621
624 622
625 623
626 624
627 625
628 626
629 627
630 628
631 629
632 630
633 631
634 632
635 633
636 634
637 635
638 636
639 637
640 638
641 639
642 640
643 641
644 642
645 643
646 644
647 645
648 646
649 647
650 648
651 649
652 650
653 651
654 652
655 653
656 654
657 655
658 656
659 657
660 658
661 659
662 660
663 661
664 662
665 663
666 664
667 665
668 666
669 667
670 668
671 669
672 670
673 671
674 672
675 673
676 674
677 675
678 676
679 677
680 678
681 679
682 680
683 681
684 682
685 683
686 684
687 685
688 686
689 687
690 688
691 689
692 690
693 691
694 692
695 693
696 694
697 695
698 696
699 697
700 698
701 699
702 700
703 701
704 702
705 703
706 704
707 705
708 706
709 707
710 708
711 709
712 710
713 711
714 712
715 713
716 714
717 715
718 716
719 717
720 718
721 719
722 720
723 721
724 722
725 723
726 724
727 725
728 726
729 727
730 728
731 729
732 730
733 731
734 732
735 733
736 734
737 735
738 736
739 737
740 738
741 739
742 740
743 741
744 742
745 743
746 744
747 745
748 746
749 747
750 748
751 749
752 750
753 751
754 752
755 753
756 754
757 755
758 756
759 757
760 758
761 759
762 760
763 761
764 762
765 763
766 764
767 765
768 766
769 767
770 768
771 769
772 770
773 771
774 772
775 773
776 774
777 775
778 776
779 777
780 778
781 779
782 780
783 781
784 782
785 783
786 784
787 785
788 786
789 787
790 788
791 789
792 790
793 791
794 792
795 793
796 794
797 795
798 796
799 797
800 798
801 799
802 800
803 801
804 802
805 803
806 804
807 805
808 806
809 807
810 808
811 809
812 810
813 811
814 812
815 813
816 814
817 815
818 816
819 817
820 818
821 819
822 820
823 821
824 822
825 823
826 824
827 825
828 826
829 827
830 828
831 829
832 830
833 831
834 832
835 833
836 834
837 835
838 836
839 837
840 838
841 839
842 840
843 841
844 842
845 843
846 844
847 845
848 846
849 847
850 848
851 849
852 850
853 851
854 852
855 853
856 854
857 855
858 856
859 857
860 858
861 859
862 860
863 861
864 862
865 863
866 864
867 865
868 866
869 867
870 868
871 869
872 870
873 871
874 872
875 873
876 874
877 875
878 876
879 877
880 878
881 879
882 880
883 881
884 882
885 883
886 884
887 885
888 886
889 887
890 888
891 889
892 890
893 891
894 892
895 893
896 894
897 895
898 896
899 897
900 898
901 899
902 900
903 901
904 902
905 903
906 904
907 905
908 906
909 907
910 908
911 909
912 910
913 911
914 912
915 913
916 914
917 915
918 916
919 917
920 918
921 919
922 920
923 921
924 922
925 923
926 924
927 925
928 926
929 927
930 928
931 929
932 930
933 931
934 932
935 933
936 934
937 935
938 936
939 937
940 938
941 939
942 940
943 941
944 942
945 943
946 944
947 945
948 946
949 947
950 948
951 949
952 950
953 951
954 952
955 953
956 954
957 955
958 956
959 957
960 958
961 959
962 960
963 961
964 962
965 963
966 964
967 965
968 966
969 967
970 968
971 969
972 970
973 971
974 972
975 973
976 974
977 975
978 976
979 977
980 978
981 979
982 980
983 981
984 982
985 983
986 984
987 985
988 986
989 987
990 988
991 989
992 990
993 991
994 992
995 993
996 994
997 995
998 996
999 997
1000 998

```

Мал.5. Створений файл README

## Завдання №4: UML діаграма

1. Підготувати діаграму створених у програмі класів та інтерфейсів за допомогою <https://app.diagrams.net/>.
2. Експортувати створену діаграму та запустити експортований файл у кореневу директорію цієї лабораторної



Мал.6. Діаграма класів

**Висновки:** я навчився дотримуватися принципів програмування та обґрунтовувати їх.

		Леус В.О.			ДУ «Житомирська політехніка».24.121.17.000 – Пр1	Арк.
		Фант М.О.				9
Змн.	Арк.	№ докум.	Підпис	Дата		