Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Факультет Информационные системы и технологии

Отчёт по лабораторной работе № 4

по дисциплине Технология программирования

Выполнил студент ИСТб-20-3 М. Д. Ярков 10.11.2021

номер группы подпись И. О. Фамилия Дата

Принял Доцент З. А. Бахвалова . .2021

должность подпись И. О. Фамилия Дата

Иркутск 2021

### Постановка задачи

### Создать иерархию классов для различных напитков. Реализовать напитки с общим свойством «объём»:

### Сок (используемый фрукт, наличие мякоти)

### Газировка (вид, количество пузыриков)

### Алкоголь (крепость, тип)

### Словесное описание алгоритма

1. При запуске программы инициализируется интерфейс, в динамический массив добавляются случайные напитки и показываются на форме с помощью метода AddDrinks\_Click
2. При нажатии кнопки «Купить напиток» вызывается метод AddDrinks\_Click:
   1. В очередь добавляется 3 случайных напитка: сок, газировка или алкоголь:
      1. При вызове статического метода GenerateRandomly классов Juice, Soda или Alcohol рандомно генерируются данные этих напитков и отдаётся объект соответствующего класса
      2. Объект добавляется в массив очереди
   2. Перерисовывается очередь с помощью метода RedrawQueue:
      1. Для каждого из 5 лейблов отображается напиток в очереди, если он существует
   3. Кнопка для покупки напитка становится активной если напиток куплен и его не забрали
3. При нажатии кнопки «Купить напиток» вызывается метод buyDrink\_Click:
   1. Последний напиток отображается в большом лейбле и удаляется из очереди
   2. Перерисовывается очередь
   3. Кнопка для получения напитка становится активной, кнопка для покупки напитка становится неактивной
4. При нажатии кнопки «Забрать напиток» вызывается метод buyDrink\_Click:
   1. Последний напиток удаляется из большого лейбла
   2. Кнопка для получения напитка становится неактивной, кнопка для покупки напитка становится активной

### Таблица спецификаций

Класс Form1: основной публичный класс, позволяющий взаимодействовать с интерфейсом. Содержит конструктор public Form1() для инициализации интерфейса при первом запуске и методы:

* private void AddDrinks\_Click(object sender, EventArgs e) – добавление напитков в очередь
* private void RedrawQueue() - перерисовка очереди
* private void BuyDrink\_Click(object sender, EventArgs e) – покупка напитка
* private void GetDrink\_Click(object sender, EventArgs e) – убор купленного напитка

Таблица 1 - поля и переменные класса Form1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля / переменной | Назначение | Тип | Диапазон |
| **Поля класса** | | | |
| helperTip | для всплывающей подсказки | объект readonly ToolTip | - |
| drinksList | динамический массив напитков внутри автомата | динамический массив List<Drink> | - |
| **private void AddDrinks\_Click(object sender, EventArgs e)** | | | |
| sender | не используется | sender | - |
| e | не используется | EventArgs | - |
| rnd | объект класса Random | Random | - |
| **private void RedrawQueue()** | | | |
| pictureBoxList | для удобного перебора лейблов | List<PictureBox> | - |
| i | итератор перебора элементов pictureBoxList | int | от -2147483648 до +2147483647 |
| **private void BuyDrink\_Click(object sender, EventArgs e)** | | | |
| переменные отсутствуют | | | |
| **private void KeyPressedHsv(object sender, KeyEventArgs e)** | | | |
| переменные отсутствуют | | | |

Класс Drinks: общий публичный класс для напитков. Содержит методы:

* public virtual String GetInfo() – получение информации о напитке
* public virtual Image GetPicture() – получение изображения напитка

Таблица 2 - поля и переменные класса Drinks

| Имя поля / переменной | Назначение | Тип | Диапазон |
| --- | --- | --- | --- |
| **Поля класса** | | | |
| rnd | объект класса Random | protected static Random | - |
| volumeInMl | объём напитка в мл | public int | от -2147483648 до 2147483647 |
| **public virtual String GetInfo()** | | | |
| переменные отсутствуют | | | |
| **public virtual Image GetPicture()** | | | |
| переменные отсутствуют | | | |

Класс Juice: публичный класс для сока, наследует Drink. Содержит методы:

* public static Juice GenerateRandomly() – получение сока с рандомными параметрами
* private String GetDescription() – получение строкового значения типа сока
* public override Image GetPicture() – получение картинки типа сока
* public override String GetInfo() – получение полной информации о соке в виде строки

Таблица 2 - поля и переменные класса Juice

| Имя поля / переменной | Назначение | Тип | Диапазон |
| --- | --- | --- | --- |
| **Поля класса** | | | |
| type | тип сока | public JuiceType | - |
| containsPulp | содержание мякоти | public bool | от -2147483648 до 2147483647 |
| **public static Soda GenerateRandomly()** | | | |
| переменные отсутствуют | | | |
| **private String GetDescription()** | | | |
| переменные отсутствуют | | | |
| **public override Image GetPicture()** | | | |
| переменные отсутствуют | | | |
| **public override String GetInfo()** | | | |
| переменные отсутствуют | | | |

Класс Soda: публичный класс для газировки, наследует Drink. Содержит методы:

* public static Soda GenerateRandomly() – получение газировки с рандомными параметрами
* private String GetDescription() – получение строкового значения типа газировки
* public override Image GetPicture() – получение картинки типа газировки
* public override String GetInfo() – получение полной информации о газировке в виде строки

Таблица 3 - поля и переменные класса Soda

| Имя поля / переменной | Назначение | Тип | Диапазон |
| --- | --- | --- | --- |
| **Поля класса** | | | |
| type | тип газировки | public SodaType | - |
| bubbleCount | количество пузыриков | public int | от -2147483648 до 2147483647 |
| **public static Soda GenerateRandomly()** | | | |
| переменные отсутствуют | | | |
| **private String GetDescription()** | | | |
| переменные отсутствуют | | | |
| **public override Image GetPicture()** | | | |
| переменные отсутствуют | | | |
| **public override String GetInfo()** | | | |
| переменные отсутствуют | | | |

Класс Alcohol: публичный класс для газировки, наследует Drink. Содержит методы:

* public static Alcohol GenerateRandomly() – получение алкоголя с рандомными параметрами
* private String GetDescription() – получение строкового значения типа алкоголя
* public override Image GetPicture() – получение картинки типа алкоголя
* public override String GetInfo() – получение полной информации об алкоголе в виде строки

Таблица 4 - поля и переменные класса Alcohol

| Имя поля / переменной | Назначение | Тип | Диапазон |
| --- | --- | --- | --- |
| **Поля класса** | | | |
| type | тип алкоголя | public AlcoholType | - |
| degree | градус крепости | public degree | от -2147483648 до 2147483647 |
| **public static Alcohol GenerateRandomly()** | | | |
| переменные отсутствуют | | | |
| **private String GetDescription()** | | | |
| переменные отсутствуют | | | |
| **public override Image GetPicture()** | | | |
| переменные отсутствуют | | | |
| **public override String GetInfo()** | | | |
| переменные отсутствуют | | | |

### Таблица тестов

| Номер теста | Что проверяем | Входные данные | Выходные данные |
| --- | --- | --- | --- |
| 1 | Заполненность автомата при запуске программы | Запуск программы | Автомат заполнен 3 случайными напитками |
| 2 | Покупка напитка | Нажатие на кнопку «Купить напиток» | Автомат переносит напиток из очереди доступных напитков в окно забора, изменяется количество доступных напитков, очередь обновляется, кнопка «Забрать напиток» становится активной, кнопка «Купить напиток» становится неактивной |
| 3 | Забор напитка, есть доступные напитки | Нажатие на кнопку «Забрать напиток» | Окно забора становится пустым, кнопка «Забрать напиток» становится неактивной, кнопка «Купить напиток» становится активной |
| 4 | Забор напитка, есть доступные напитки | Нажатие на кнопку «Забрать напиток» | Окно забора становится пустым, кнопка «Забрать напиток» становится неактивной, кнопка «Купить напиток» остаётся неактивной |
| 5 | Пополнение автомата, в окне забора нет напитка | Нажатие на кнопку «Пополнить автомат» | Очередь и количество доступных напитков обновляются, кнопка «Купить напиток» становится активной |
| 6 | Пополнение автомата, в окне забора есть напиток | Нажатие на кнопку «Пополнить автомат» | Очередь и количество доступных напитков обновляются, кнопка «Купить напиток» остаётся неактивной |
| 7 | Автотест 1: вывод правильной информации об алкоголе | Первая проверка:  600 мл  пиво  4 градуса  Вторая проверка:  600 мл  Ягуар  8 градусов  Третья проверка:  600 мл  виски  40 градусов | Первый вывод:  Алкоголь (пиво, объём 600 мл, крепость 4 градусов)  Второй вывод:  Алкоголь (ягуар, объём 600 мл, крепость 8 градусов)  Третий вывод:  Алкоголь (виски, объём 600 мл, крепость 40 градусов) |
| 8 | Автотест 2: вывод правильной информации о газировке | Первая проверка:  600 мл  Байкал  135 пузырьков Вторая проверка:  600 мл  Буратино  135 пузырьков Третья проверка:  600 мл  Фанта  135 пузырьков | Первый вывод:  Газировка "Байкал" (объём 600 мл, содержит 135 пузырьков)  Второй вывод:  Газировка "Буратино" (объём 600 мл, содержит 135 пузырьков)  Третий вывод:  Газировка "Фанта" (объём 600 мл, содержит 135 пузырьков) |
| 9 | Автотест 3: вывод правильной информации о соке | Первая проверка:  500 мл  яблочный  не содержит мякоть  Вторая проверка:  500 мл  виноградный  содержит мякоть  Третья проверка:  500 мл  апельсиновый  содержит мякоть | Первый вывод:  Яблочный сок (объём 500 мл, не содержит мякоть)  Второй вывод:  Виноградный сок (объём 500 мл, содержит мякоть)  Третий вывод:  Апельсиновый сок (объём 500 мл, содержит мякоть) |

### Код программы

Код программы находится по ссылке https://github.com/mxkmn/RepoForMyTeacher\_CSharpEdition/tree/main/Lab4Var1

Файл с автотестами:

|  |
| --- |
| using Microsoft.VisualStudio.TestTools.UnitTesting;  namespace Lab4Var1.Tests {  [TestClass()]  public class AlcoholTests {  [TestMethod()]  public void GetInfoTest() {  bool isCorrect = true;  Alcohol drink = new Alcohol();  drink.volumeInMl = 600;  drink.type = AlcoholType.beer;  drink.degree = 4;  if (drink.GetInfo() != "Алкоголь (пиво, объём 600 мл, крепость 4 градусов)")  isCorrect = false;  drink.type = AlcoholType.jaguar;  drink.degree = 8;  if (drink.GetInfo() != "Алкоголь (ягуар, объём 600 мл, крепость 8 градусов)")  isCorrect = false;  drink.type = AlcoholType.whiskey;  drink.degree = 40;  if (drink.GetInfo() != "Алкоголь (виски, объём 600 мл, крепость 40 градусов)")  isCorrect = false;  Assert.IsTrue(isCorrect);  }  }  [TestClass()]  public class SodaTests {  [TestMethod()]  public void GetInfoTest() {  bool isCorrect = true;  Soda drink = new Soda();  drink.volumeInMl = 600;  drink.type = SodaType.baikal;  drink.bubbleCount = 135;  if (drink.GetInfo() != "Газировка \"Байкал\" (объём 600 мл, содержит 135 пузырьков)")  isCorrect = false;  drink.type = SodaType.buratino;  if (drink.GetInfo() != "Газировка \"Буратино\" (объём 600 мл, содержит 135 пузырьков)")  isCorrect = false;  drink.type = SodaType.fanta;  if (drink.GetInfo() != "Газировка \"Фанта\" (объём 600 мл, содержит 135 пузырьков)")  isCorrect = false;  Assert.IsTrue(isCorrect);  }  }  [TestClass()]  public class JuiceTests {  [TestMethod()]  public void GetInfoTest() {  bool isCorrect = true;  Juice drink = new Juice();  drink.volumeInMl = 500;  drink.type = JuiceType.apple;  drink.containsPulp = false;  if (drink.GetInfo() != "Яблочный сок (объём 500 мл, не содержит мякоть)")  isCorrect = false;  drink.type = JuiceType.grape;  drink.containsPulp = true;  if (drink.GetInfo() != "Виноградный сок (объём 500 мл, содержит мякоть)")  isCorrect = false;  drink.type = JuiceType.orange;  if (drink.GetInfo() != "Апельсиновый сок (объём 500 мл, содержит мякоть)")  isCorrect = false;  Assert.IsTrue(isCorrect);  }  }  } |

### Тесты



Рисунок 1 – тест 1



Рисунок 2 – тест 2



Рисунок 3 – тест 3

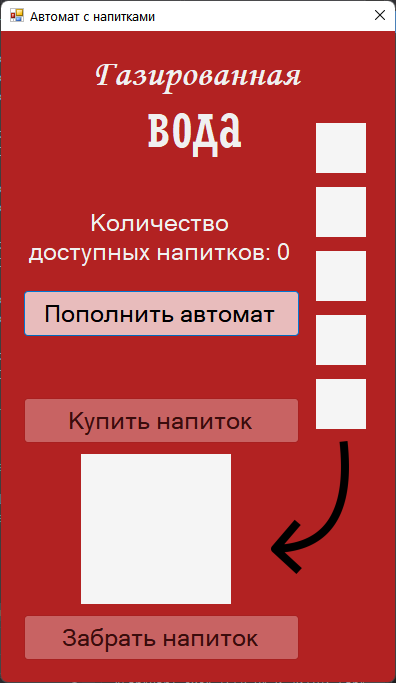


Рисунок 4 – тест 4



Рисунок 5 – тест 5



Рисунок 6 – тест 6