Aplikacja konsolowa jak na egzaminie — parking

Czas: 2 × 45 min

praca samodzielna na czas — warunki egzaminacyjne (dozwolona tylko dokumentacja Microsoft Learn)

Wykład 1.13–1.21

Cel: W 90 minut wykonasz zadanie w stylu części konsolowej egzaminu INF.04: klasę z polem statycznym i walidacją, program główny, testy jednostkowe i dokumentację ze zrzutami.

Po zajęciach potrafisz:

- realizuje specyfikację klasy w wyznaczonym czasie
- pisze program obsługujący błędne dane wejściowe
- tworzy testy jednostkowe dla wartości granicznych
- przygotowuje dokumentację kodu i zrzuty ekranu jak na egzaminie

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.

Przebieg zadania

1 Specyfikacja klasy Bilet

Utwórz rozwiązanie `Parking` z aplikacją konsolową, biblioteką klas i projektem MSTest (jak w zadaniu 22).

1. Statyczna właściwość `LiczbaBiletow` (tylko do odczytu z zewnątrz) — liczba wydanych biletów.
2. Właściwość `Numer` — kolejny numer biletu nadawany w konstruktorze (1, 2, 3...).
3. Właściwość `Minuty` — czas postoju podany w konstruktorze; dla wartości spoza 0–10080 konstruktor zgłasza `ArgumentOutOfRangeException`.
4. Metoda `decimal Oplata()`: do 15 minut włącznie — 0 zł; powyżej — 3 zł za każdą rozpoczętą godzinę liczoną od początku postoju, najwyżej 40 zł za dobę; za każdą pełną dobę 40 zł, a pozostałe minuty liczone jak wyżej (z limitem 40 zł).

2 Program główny

Kasa parkingu.

1. Program pyta o czas postoju w minutach, aż użytkownik wpisze pusty wiersz.
2. Dla każdego poprawnego czasu wypisuje: `Bilet nr 1: 90 min, do zapłaty 6,00 zł`.
3. Dla tekstu niebędącego liczbą i czasu spoza zakresu wypisuje zrozumiały komunikat i pyta ponownie.
4. Na końcu wypisuje liczbę wydanych biletów i utarg.

3 Testy jednostkowe

Co najmniej 10 przypadków dla metody `Oplata` i testy wyjątków.

1. Przed pisaniem testów oblicz ręcznie oczekiwane opłaty dla: 0, 15, 16, 60, 61, 780, 781, 1440, 1441 i 3000 minut.
2. Zapisz je jako jeden test z `[DataRow]`.
3. Dodaj testy wyjątku dla −1 i 10081 oraz test kolejnych numerów biletów.
4. Jeśli test nie przechodzi — sprawdź najpierw, czy dobrze policzyłeś oczekiwany wynik, potem kod.

4 Dokumentacja

Jak na egzaminie.

1. Komentarze `///` przy klasie, konstruktorze, właściwościach i metodzie (opis, parametry, wynik, wyjątki).
2. W nagłówku plików wpisz zamiast nazwiska numer (np. numer z dziennika) — na egzaminie podaje się numer PESEL lub numer zdającego.
3. Zrzuty ekranu: działanie programu (poprawne i błędne dane) oraz wynik testów w Eksploratorze testów.
4. Spakuj folder rozwiązania (bez `bin` i `obj`) i oddaj zgodnie z poleceniem nauczyciela.

W sprawozdaniu umieść

- ☐ kod klasy Bilet z dokumentacją
- ☐ kod programu głównego
- ☐ tabela ręcznych obliczeń opłat
- ☐ zrzuty: program i Eksplorator testów
- ☐ czas wykonania każdej części (samoocena)

Pytania kontrolne

1. Która część zadania zajęła najwięcej czasu i jak możesz to przyspieszyć na egzaminie?

2. Dlaczego numer biletu można nadać na podstawie licznika statycznego?
3. Jakie wartości graniczne ma metoda Oplata i dlaczego to one są najważniejsze w testach?

Kryteria oceny

Kryterium	Punkty
Klasa Bilet zgodna ze specyfikacją	6
Program główny z obsługą błędów	4
Testy (min. 10 przypadków, wyjątki)	4
Dokumentacja /// i zrzuty	4
Organizacja rozwiązania i pytania	2
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).