

## **Programowanie w języku Java (projekt)**

**Temat nr 24 Aplikacja(standalone) do prowadzenia oraz zarządzania sklepem komputerowym**

### **Grupa 2ID13A**

Grzegorz Pedycz  
Pavlo-Jan Dediu  
Witkor Kondraciuk

**25.06.2021**

W danym projekcie została użyta baza PostgreSQL wersji 42.2.19 oraz biblioteka javafx wersji 13,  
W zespole opracowaliśmy funkcjonalność projektu ze strony klientowej, czyli zostały dodane takie funkcjonalności jak

- możliwość logowania oraz tworzenia kont dla użytkowników (klientów)
- podział sprzętu na kategorie ,karty graficzne i tp
- dodawanie sprzętu do koszyka oraz sprzedaż

### **tabele oraz przykładowe insert'y do bazy danych**

-----

--tabela klient będzie przechowywała konta użytkowników

```
CREATE TABLE public."Klient"  
(  
    "Id_klient" integer NOT NULL,  
    "firstName" text COLLATE pg_catalog."default" NOT NULL,  
    "surName" text COLLATE pg_catalog."default" NOT NULL,  
    "userName" text COLLATE pg_catalog."default" NOT NULL,  
    password text COLLATE pg_catalog."default" NOT NULL,  
    "saldo" integer NOT NULL,          --update 0.1  
    CONSTRAINT "Klient_pkey" PRIMARY KEY ("Id_klient")  
)
```

--tabela Koszyk będzie przechowywała wybrane przez użytkownika (klienta) produkty

```
CREATE TABLE public."Koszyk"  
(  
    "Id_koszyk" integer NOT NULL,  
    "Id_klient" integer NOT NULL,  
    "Id_produkt" integer NOT NULL,  
    "Cena" integer,  
    CONSTRAINT "Koszyk_pkey" PRIMARY KEY ("Id_koszyk"),  
    CONSTRAINT "Klient_Koszyk_fkey" FOREIGN KEY ("Id_klient") REFERENCES  
public."Klient"("Id_klient"),  
    CONSTRAINT "Produkt_Koszyk_fkey" FOREIGN KEY ("Id_produkt") REFERENCES  
public."Produkt"("Id_produkt")  
)
```

--tabela Produkt będzie przechowywała wszystkie produkty które są w sklepie

```
CREATE TABLE public."Produkt"  
(  
    "Id_produkt" integer NOT NULL,  
    "Nazwa" text COLLATE pg_catalog."default" NOT NULL,  
    "Cena" integer NOT NULL,  
    "Kategoria" text COLLATE pg_catalog."default" NOT NULL,  
    "Typ" text COLLATE pg_catalog."default" NOT NULL,  
    CONSTRAINT "Produkt_pkey" PRIMARY KEY ("Id_produkt")  
)
```

```

insert into public."Produkt" values(1, 'ASUS GeForce RTX 3080 Ti ROG STRIX 12GB', 9590,
'Komputery', 'Karty graficzne');
insert into public."Produkt" values(2, 'ASUS GeForce RTX 3070 Ti ROG STRIX 8GB', 5490,
'Komputery', 'Karty graficzne');
insert into public."Produkt" values(3, 'Gigabyte GeForce GTX 1660 Ti 6GB', 2149, 'Komputery',
'Karty graficzne');
insert into public."Produkt" values(4, 'ASUS TUF GAMING B550-PLUS', 599, 'Komputery', 'Płyty
główne');
insert into public."Produkt" values(5, 'ASUS Prime Z490-A', 1159, 'Komputery', 'Płyty główne');
insert into public."Produkt" values(6, 'Corsair CV 550W CP-9020210-EU', 209, 'Komputery',
'Zasilacze do komputera');
insert into public."Produkt" values(7, 'Chieftec BDF-500S', 249, 'Komputery', 'Zasilacze do
komputera');
insert into public."Produkt" values(8, 'Crucial Ballistix RGB White 16GB [2x8GB 3200MHz DDR4
CL16 DIMM]', 449, 'Komputery', 'Pamięci RAM');
insert into public."Produkt" values(9, 'HyperX Fury Black 32GB [2x16GB 2666MHz DDR4 CL16 XMP
1.2V DIMM]', 779, 'Komputery', 'Pamięci RAM');
insert into public."Produkt" values(10, 'AMD Ryzen 5 3600', 919, 'Komputery', 'Procesory');
insert into public."Produkt" values(11, 'Intel Core i7-10700K', 1479, 'Komputery', 'Procesory');

insert into public."Produkt" values(12, 'Packard Bell NEW91 TM85', 399, 'Laptopy', 'Płyty główne
lp');
insert into public."Produkt" values(13, 'Acer LA-5912P Rev:1.0', 249, 'Laptopy', 'Płyty główne
lp');
insert into public."Produkt" values(14, 'LA-9591P Dell Latitude E7440', 310, 'Laptopy', 'Płyty
główne lp');
insert into public."Produkt" values(15, 'GOODRAM 8GB (1x8GB) 1600MHz CL11', 170,
'Laptopy', 'Pamięci RAM lp');
insert into public."Produkt" values(16, 'GOODRAM 16GB (1x16GB) 3200MHz CL22', 340,
'Laptopy', 'Pamięci RAM lp');
insert into public."Produkt" values(17, 'Intel Core i5 520M 2,4GHz', 320, 'Laptopy', 'Procesory
lp');
insert into public."Produkt" values(18, 'Intel Core i7-3540M 3.0-3.7GHz SR0X6', 370,
'Laptopy', 'Procesory lp');

```

-----

## klasa główna App.java

-----

```
//Klasa główna dziedzicząca
```

-----

## Configs.java

-----

```
//w danej klasie zostały zdefiniowane stałe które służą do przechowywania danych wejściowych dla
połączenia z bazą
```

-----

## User.java

-----

```
//dana klasa odwzorowujea użytkownika(klienta)
//w danej klasie zostały zdefiniowane metody Get i Set do podrania pól opisowych klasy
```

-----

## Produkt.java

```
-----  
//klasa odwzorowująca produkt w sklepie  
//w danej klasie zostały zdefiniowe metody Get i Set do podrania pól opisowych klasy  
-----
```

## UserSession.java

```
-----  
//klasa singleton dająca możliwość przesyłać dane klienta innym klasom w ramach jego sesji  
//w danej klasie zostały zdefiniowe metody Get i Set do podrania pól opisowych klasy  
-----
```

## LoginController.java

```
-----  
//klasa odpowiadająca za logowanie  
//w danej klasie zostały zdefiniowane metody:  
//initialize inicjalizująca elementy javafx,  
//LoginUser która sprawdza hasło u nazwę użytkownika w bazie,  
//openNewScene metoda służąca do przejścia na nowe okno.
```

## SignupController.java

```
-----  
//klasa odpowiadająca za tworzenia konta użytkownika  
//w danej klasie zostały zdefiniowane metody:  
//initialize inicjalizująca elementy javafx,  
//signUpNewUser metoda która wykonuje utworzenie konta dla nowego użytkownika  
//openNewScene metoda służąca do przejścia na nowe okno.  
-----
```

Store manager  
Winley v0.1

Sign Up

Authentication

Username

Password

Log In

Store manager  
Winley v0.1

First Name

John

Surname

Anderson

Username

neo

●●●

●●●

Password

Re-type password

Account created successfully

Back

Confirm

Store manager  
Winley v0.1

First Name

Jan

Surname

Kowalski

Username

neo

●●●●

●●●●

Password

Re-type password

This username exists, you must put another

Back

Confirm

## MainMenu.java

```
-----  
//klasa służąca do inicjalizacji głównego menu  
//w danej klasie zostały zdefiniowane metody:  
//initialize metoda inicjalizująca elementy javafx,  
//handleMouseClicked metoda która przepisuje nazwę wybranego myszką węzła w drzewie TreeView do  
zmiennej,  
//printResultSearch metoda która poszukuje produkt według słowa kluczowego podanego jako  
argument  
//printBalance metoda wyświetlająca ewentualne środki na koncie  
//openNewScene metoda służąca do przejścia na nowe okno.  
-----
```

KOSZYK

WYLOGUJ

# Welcome in Store Manager

Kategoria	Nazwa	Typ	Cena
No content in table			

Dodaj

## Choose department

► Kategorie

Zatwierdź

KOSZYK

WYLOGUJ

# Welcome in Store Manager

[illegible]

Dodaj

## Choose department

▼ Kategorie

▼ Komputery

## Karty graficzne

## Płyty główne

## Zasilacze do komputera

Pamięci RAM

## Procesory

Laptopy

- ▶ Laptopy

Zatwierdź

KOSZYK

WYLOGUJ

## Welcome in Store Manager

[illegible]

Dodaj

## Choose department

▼ Kategorie

▼ Komputery

## Karty graficzne

## Płyty główne

## Zasilacze do komputera

Pamięci RAM

## Procesory

▼ Laptopy

## Płyty główne 1p

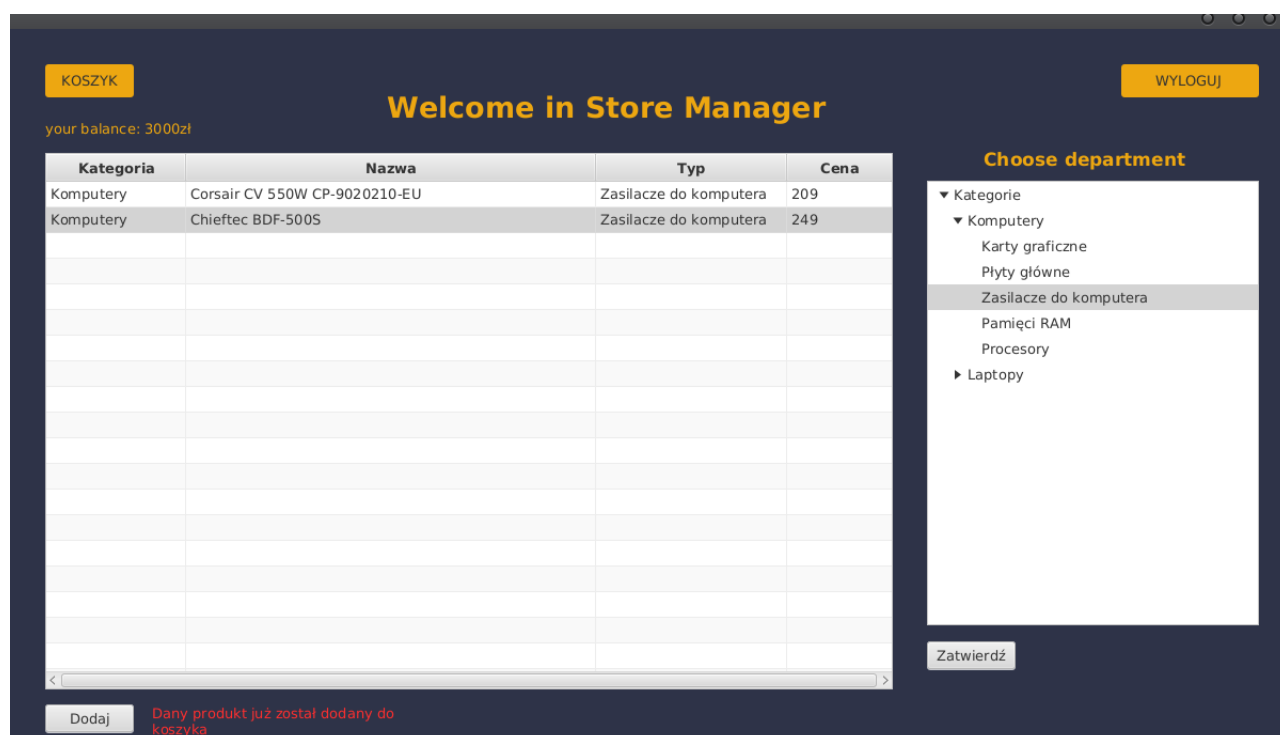
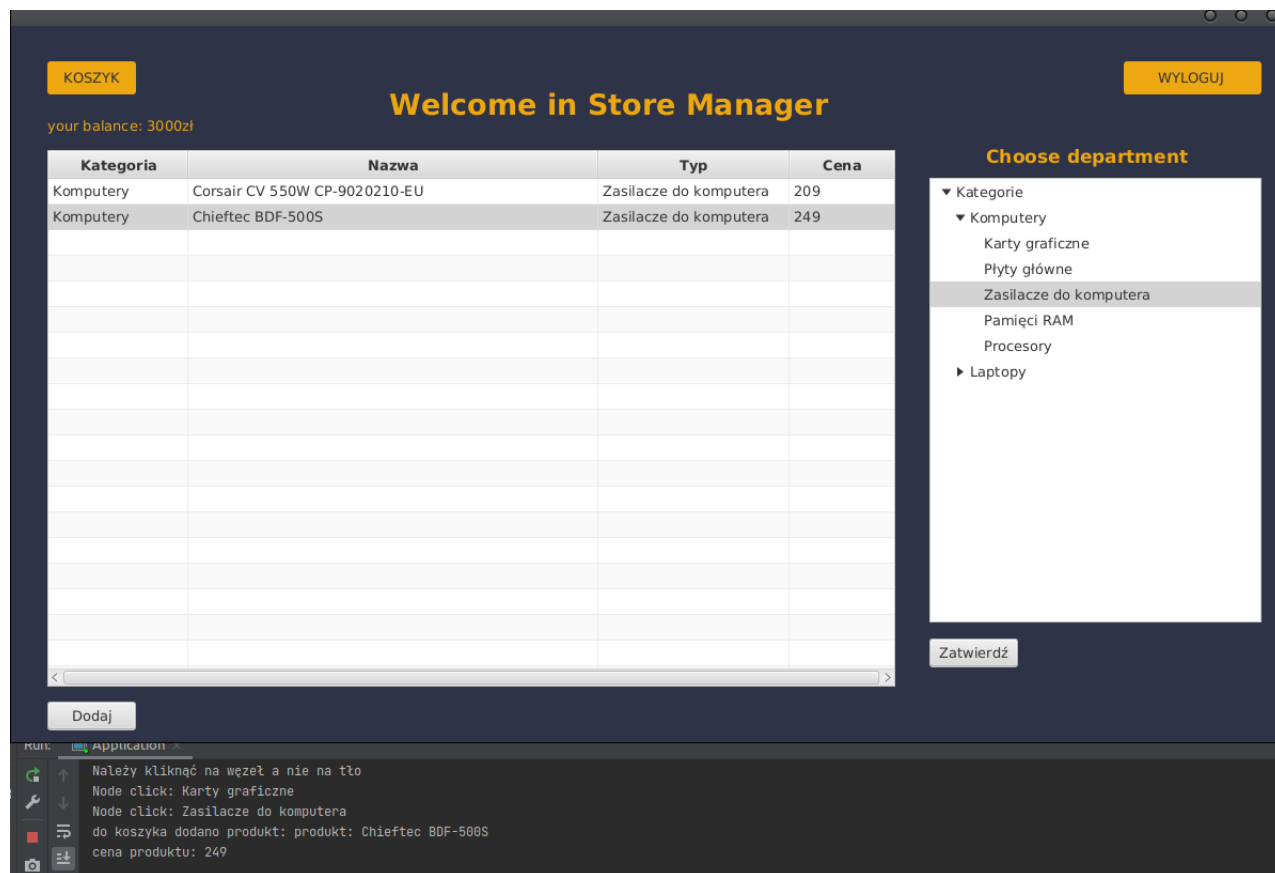
Pamięci RAM Ip

### Procesory I/O

Zatwierdź

## Koszyk.java

```
//klasa odwzorowująca koszyk użytkownika(klienta)
//w danej klasie zostały zdefiniowane metody:
//initialize metoda inicjalizująca elementy javafx,
//printBalance metoda wyświetlająca ewentualne środki na koncie
//openNewScene metoda służąca do przejścia na nowe okno.
```





## DatabaseHandler.java

```
/*
w tej klasie podłączamy naszą bazę danych
do aplikacji oraz definiujemy różną działalność z nią
*/
//w danej klasie zostały zdefiniowane metody:
//getConnection metoda opisująca nawiązywanie połączenia z bazą
oraz zwracająca obiekt klasy Connection,
//signUpUser metoda zapisująca użytkownika do bazy
//getNewUser_id metoda zwracająca nowy identyfikator (stary identyfikator + 1)
//getCurrentUser_id metoda zwracająca identyfikator użytkownika która wyszukuje ten
identyfikator za nazwą użytkownika podaną w argumencie
//getUser metoda która służy do poszukiwania użytkowników w bazie, zwraca rekord z bazy,
przyjmuje obiekt klasy user jako argument
//getAllProducts metoda zwracająca liste produktów które są pogrupowane według słowa kluczowego
podanego w argumencie
//setKoszyk metoda służąca do dodania produktu do koszyka, przyjmuje obiekt klasy produkt jako
argument
//getKoszyk metoda zwracająca liste koszyków do których użytkownik dodaje produkty
//getTotalPrice metoda zwracająca sumę należną do zapłaty dla ewentualnego użytkownika
//buyProducts metoda służąca do zakupu produktów z koszyka
//deleteItem metoda służąca do usuwania produktu z koszyka, przyjmuje argument obiekt klasy
Koszyk
//getBalance metodą zwracająca ewentualne środki na koncie ewentualnego użytkownika
```



## Twoje zakupy

[POWRÓT](#)

USUÑ

ZATWIERDŹ

[illegible]

9839zł

## Twoje zakupy

[POWRÓT](#)

USUÑ

ZATWIERDŹ

[illegible]

698zł