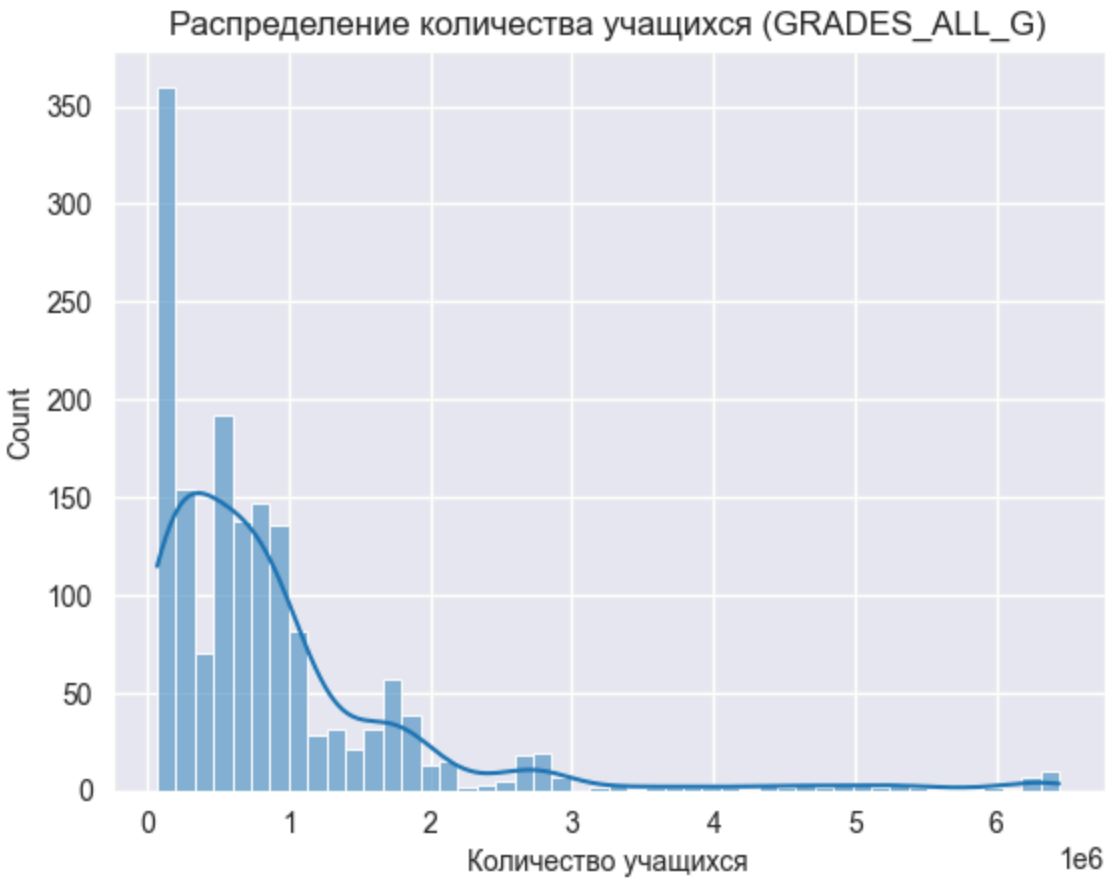
Пичурин Вадим РТ5-61Б

РК2 ТМО

In [2]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.tree import DecisionTreeRegressor
from xgboost import XGBRegressor

# Загрузка данных
df = pd.read_csv('data/states_all.csv')

# Целевая переменная — общее количество учеников
target = 'GRADES_ALL_G'

# Удаляем строки с отсутствующим target
df = df[df[target].notna()]

# Удаляем признаки, бесполезные для регрессии
df = df.drop(columns=['PRIMARY_KEY', 'STATE'])

# Разделяем на признаки и целевую переменную
X = df.drop(columns=[target])
y = df[target]

# Заполняем пропуски средним
imputer = SimpleImputer(strategy='mean')
X_imputed = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

# Масштабируем числовые признаки
scaler = StandardScaler()
X_scaled = pd.DataFrame(scaler.fit_transform(X_imputed), columns=X_imputed.col

# Делим на train/test
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2
```

In [6]:
```
import matplotlib.pyplot as plt
import seaborn as sns

sns.histplot(y, kde=True)
plt.title("Распределение количества учащихся (GRADES_ALL_G)")
plt.xlabel("Количество учащихся")
plt.show()
```



In [3]:
```
tree_model = DecisionTreeRegressor(random_state=42)
tree_model.fit(X_train, y_train)
tree_preds = tree_model.predict(X_test)
```

In [4]:
```
boost_model = XGBRegressor(random_state=42)
boost_model.fit(X_train, y_train)
boost_preds = boost_model.predict(X_test)
```

In [5]:
```
def evaluate_model(name, y_true, y_pred):
    print(f"{name}:")
    print("MSE:", mean_squared_error(y_true, y_pred))
    print("R2:", r2_score(y_true, y_pred), "\n")

evaluate_model("Decision Tree", y_test, tree_preds)
evaluate_model("Gradient Boosting", y_test, boost_preds)
```

```
Decision Tree:
MSE: 3584679436.825688
R2: 0.9972486107688624

Gradient Boosting:
MSE: 1299177247.7475786
R2: 0.9990028279092211
```